



Silk Test 21.0

Silk Test Workbench ヘルプ

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
<http://www.microfocus.com>

© Copyright 1992-2021 Micro Focus またはその関連会社。

MICRO FOCUS、Micro Focus のロゴおよび Silk Test は、Micro Focus またはその関連会社の商標または登録商標です。

その他、記載の各名称は、各所有社の知的所有財産です。

2021-09-13

目次

Silk Test Workbench 21.0 へようこそ	23
ライセンス情報	25
はじめに	26
ログオン	26
ログオン パスワードの変更	26
Silk Test Workbench の概要	27
Silk Test Workbench を使用する利点	27
特殊な状況下での自動化 (周辺機器が無い)	28
Silk Test 製品スイート	29
Silk Test Workbench の機能	30
Silk Test Workbench の UI	31
Silk Test Workbench ソフトウェア コンポーネント	32
自動テストについて	32
テスト戦略	32
別の場所にあるデータベースの利用について	33
Silk Test Workbench ドキュメント	34
Silk Test Workbench の新機能	34
HTML 形式の結果レポートにおけるイメージ検証	34
ブラウザ サポートの自動更新	34
スタンドアロン Recorder	34
AutoPass のサポート	35
ユーザビリティの改善	35
使用技術の更新	35
プロジェクトの管理	35
プロジェクトの表示	36
ショートカット キー	36
チュートリアル	38
Silk Test Workbench ビジュアル テスト チュートリアル	38
Silk Test Workbench スクリプト チュートリアル	67
ヘルプについてのヘルプ	75
ヘルプを表示する方法	75
ヘルプ トピックの印刷	76
使用状況データの収集の有効化/無効化	76
Micro Focus へのお問い合わせ	76
Micro Focus SupportLine が必要とする情報	77
テストの作成	78
ビジュアル テストの作成	78
ビジュアル テストの記録	78
開始画面からのビジュアル テストの記録	80
複数のテスト アプリケーションをテストするビジュアル テストの記録	82
ビジュアル テストでのロケータからオブジェクト マップ エントリへの移動	83
Java Network Launching Protocol (JNLP) を使用するアプリケーションを起動するビジュアル テスト	83
新しいブラウザ ウィンドウで開くビジュアル テストの構成	84
ビジュアル テスト内での特定のステップの表示	85
テスト アプリケーションでのビジュアル テスト用コントロールの識別	85
画面プレビューでのビジュアル テスト用コントロールの識別	86
オブジェクトの識別ダイアログを使用したビジュアル テスト用コントロールの識別	86

特定のテストステップでの追加操作の記録	87
テストアプリケーションからビジュアルテストへの画面の挿入	88
ビジュアルテストのオブジェクト マップ項目またはロケーターの手動による記録	88
既存のビジュアルテストを開く	89
ビジュアルテストの保存	90
ビジュアルテストの概要	90
プロパティ	99
スクリプトの作成	109
スクリプトを使用する利点	110
テストスクリプト作成のベストプラクティス	110
スクリプトの構文	111
ドライバスクリプト	112
手動によるスクリプト作成	113
スクリプトの記録	114
複数のテストアプリケーションをテストするスクリプトの記録	116
Java Network Launching Protocol (JNLP) を使用するアプリケーションを起動するスクリプトの	117
新しいブラウザウィンドウで開くスクリプトの構成	117
テストアプリケーションでのコントロールの識別 (.NET スクリプト)	118
スクリプト内からのアプリケーションの起動	119
スクリプトのオブジェクト マップ項目またはロケーターの手動による記録	120
既存のテストへの追加操作の記録	121
既存のスクリプトを開く	121
スクリプトの編集	121
スクリプトの保存	122
スクリプトでの要素の定義への移動	122
アプリケーション構成	122
アプリケーション構成の変更	123
[アプリケーションの選択] ダイアログ ボックス	125
リモートロケーションの編集	125
ユーザーアカウント制御	126
特定のテクノロジドメインの無効化	126
アプリケーション構成エラー	127
アプリケーション構成のトラブルシューティング	128
アプリケーション構成での環境変数	128
基本状態	128
基本状態のオン/オフの切り替え	129
基本状態の実行	129
ユーザーインターフェイスからの基本状態の変更	130
スクリプトでの基本状態の変更	132
記録の概要	134
記録中のオブジェクトのハイライト	134
記録中および再生中に除外される文字	134
記録中に利用可能なアクション	134
テストの拡張	136
アクティブデータ	136
アクティブデータテストの利点	137
アクティブデータ資産	137
アクティブデータテスト資産の作成	137
アクティブデータテストのためのテストファイルの準備	138
アクティブデータテスト用の新しいデータファイルの作成	139
アクティブデータテストファイルに対する読み取りオプションの設定	140
アクティブデータテストのためのアクティブデータファイルの編集	141
アクティブデータに使用するデータの指定	141

ビジュアルテストでのアクティブ データの使用	143
スクリプトでのアクティブ データの使用	150
ビジュアル テストの拡張	153
ビジュアル テスト ステップで実行される操作の変更	153
ビジュアル テストへのコメントの挿入	153
フラグ	154
テスト ロジックの追加	156
自動ステップの操作	186
ビジュアル テストでの項目識別子の使用	191
ビジュアル テストでのテスト データの再利用	191
テスト フローの管理	207
ビジュアル テストの比較	216
ビジュアル テストのマージ	217
スクリプトの拡張	218
スクリプト内からのスクリプトの参照	218
記録中のスクリプトへの検証の追加	219
コーディングによるスクリプトへの検証の追加	219
スクリプトでのテスト データの再利用	220
スクリプトの管理	235
Windows DLL の呼び出し	241
スクリプトからの Windows DLL の呼び出し	242
DLL 関数の宣言構文	242
DLL 呼び出しの例	243
DLL 関数への引数の受け渡し	243
DLL 関数で変更できる引数の受け渡し	244
DLL 関数への文字列引数の受け渡し	245
DLL 名のエイリアス設定	245
DLL 関数呼び出しの表記規則	246
Microsoft ユーザー補助を使用したオブジェクト解決の向上	246
ユーザー補助の使用	247
ユーザー補助の有効化	247
Microsoft UI オートメーション	247
UI オートメーション プロバイダー インターフェイスを実装したアプリケーションに対するテスト	248
UI オートメーション メソッドの動的呼び出し	249
UI オートメーションを使ってコントロールを識別するためのロケーター属性	251
UI オートメーション コントロールでのスクロール	252
UI オートメーション使用時の制限事項	253
UI オートメーション サポートを使ったテスト時のトラブルシューティング	253
Silk Test Workbench での Unicode コンテンツのサポートの概要	254
テキスト解決のサポート	254
カスタム コントロール	256
動的呼び出し	256
テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする	257
Apache Flex カスタム コントロールのテスト	260
カスタム コントロールの管理	261
実行時間の計測	264
テスト実行の遅延	264
パスワードの暗号化	265
テストのデバッグ	266
ビジュアル テストのデバッグ	266
選択したポイントでのビジュアル テスト再生の停止	266
選択したポイントからのビジュアル テスト再生の段階的実行	266
ビジュアル テストの再生におけるステップ実行の制御	267

デバッグ中のビジュアル テストの編集	267
デバッグ モードでのビジュアル テストの段階的実行	268
ビジュアル テスト内でのエラー処理による確実な再生	268
ビジュアル テストの印刷	270
スクリプトのデバッグ	271
スクリプトの信頼性	272
デバッグ モードでのスクリプトの段階的実行	272
選択したポイントからのスクリプト再生の段階的実行	273
スクリプト デバッグ中の行の実行の制御	273
スクリプト例 - エラー処理	273
スクリプトへのコメントの挿入	274
選択したポイントでのスクリプト再生の停止	274
スクリプトの印刷	275
テストの再生と結果の分析	276
ビジュアル テストの再生	276
ビジュアル テストの再生	276
ビジュアル テスト内でのビジュアル テストの実行	277
[ビジュアル テスト変数] ダイアログ ボックス	278
スクリプトの再生	279
スクリプトの再生	279
再生中のオートメーション操作の表示	280
テストの並列実行	280
ビジュアル テスト再生の構成	281
ビジュアル テスト内での再生設定値の設定	282
ビジュアル テスト内での再生設定値の取得	282
挿入されたビジュアル テストまたは VB .NET スクリプトの再生の構成	283
Silk Test Workbench がテストを同期する方法	283
結果の分析	285
[結果] ウィンドウ	286
結果の管理	291
結果フィルタ	297
[条件] ダイアログ ボックス	300
再生ステータス ダイアログ ボックスの有効化	300
[再生完了] ダイアログ ボックス	301
[再生エラー] ダイアログ ボックス	301
資産の管理	302
資産の種類	302
資産のバージョンをパージ	303
資産のバージョンをパージ	303
パージのコマンド ライン パラメータ	304
アセット ブラウザを使用した資産の作成	307
スクリプトから資産を開く	307
資産を開く	307
資産の比較	307
資産のマーシ	308
資産の複製	309
資産名の変更	309
資産の削除	310
資産リストのフィルタ処理	310
資産の検索	311
資産の並べ替え	311
資産保存時のデフォルト動作の設定	311
資産のバージョンの最大数の設定	311
インクルードする資産の表示	312
複数バージョンの資産の表示	312

資産の命名規則	313
テスト資産のインポートとエクスポート	313
資産のインポート	313
資産のエクスポート	315
資産のインポートとエクスポートの権限	316
オブジェクト解決	317
ロケーターの基本概念	317
オブジェクト タイプと検索範囲	317
属性を使用したオブジェクトの識別	318
ロケーターの構文	318
スクリプトでのロケーターの使用	320
Find メソッドの使用	321
ロケーターを使用したオブジェクトの存在確認	321
1 つのロケーターで複数のオブジェクトを識別する	322
ロケーターのカスタマイズ	322
安定した識別子	322
カスタム属性	325
XPath のパフォーマンス問題のトラブルシューティング	327
Silk Test Open Agent	329
Silk Test Open Agent の起動	329
テスト実行後に Open Agent を停止する	329
エージェント オプション	329
Silk Test Workbench コンポーネント間の接続設定	338
Open Agent に接続するポートの構成	339
Information Service に接続するポートの構成	340
Silk Test Information Service プロパティの編集	341
Information Service との HTTPS 接続に使用する証明書の置換	342
Open Agent を使用したリモートテスト	343
リモート Open Agent を使用したテスト	343
オブジェクト マップ	344
オブジェクト マップを使用する利点	345
オブジェクト マップのオン/オフの切り替え	345
オブジェクト マップのスコープ	345
操作の記録中でのオブジェクト マップのマージ	346
Web アプリケーションでのオブジェクト マップの使用	347
オブジェクト マップ項目名の変更	347
オブジェクト マップの変更	349
オブジェクト マップのロケーターの変更	350
オブジェクト マップの比較	351
オブジェクト マップのマージ	352
テスト アプリケーションからのオブジェクト マップの更新	352
オブジェクト マップ項目のコピー	354
オブジェクト マップ項目の追加	355
スクリプトからオブジェクト マップを開く	356
テスト アプリケーションでのオブジェクト マップ項目の検索	357
オブジェクト マップのエラーの検出	358
オブジェクト マップ項目の削除	359
オブジェクト マップを最初に書き出す	359
オブジェクト マップの要素のグループ化	360
オブジェクト マップ：よくある質問	361
複数のオブジェクト マップを単一のオブジェクト マップのマージする方法	361
テスト スクリプトを削除したときにオブジェクト マップで起こること	361
テスト対象アプリケーションのオブジェクト マップを手動で作成する方法	361
イメージ解決のサポート	362

イメージ クリックの記録	362
イメージ解決メソッド	362
イメージ資産	363
イメージ資産の作成	363
同じイメージ資産に複数のイメージを追加する	364
スクリプトから資産を開く	365
イメージ クリックの検証資産名での変数の使用	365
イメージ検証	365
イメージ検証の作成	365
ビジュアル テストにイメージ検証を追加する	366
記録中にイメージ検証を追加する	367
イメージ検証が失敗した理由の確認	367
イメージ検証の検証資産名での変数の使用	367
イメージ検証の相違点ビュー	368
特定の環境のテスト	369
Apache Flex のサポート	369
Adobe Flash Player で実行するための Flex アプリケーションの構成	369
Component Explorer の起動	370
Apache Flex アプリケーションのテスト	370
Apache Flex カスタム コントロールのテスト	370
Apache Flex スクリプトのカスタマイズ	380
同一 Web ページ上の複数の Flex アプリケーションのテスト	380
Adobe AIR のサポート	381
名前またはインデックスを使用する Flex の Select メソッドの概要	381
FlexDataGrid コントロールでの項目の選択	382
Flex アプリケーションのテストの有効化	382
Apache Flex アプリケーションのスタイル	394
Adobe Flash Player のセキュリティ制約に対応するための Flex アプリケーションの構成	395
Apache Flex アプリケーションの属性	395
Silk Test Workbench が Apache Flex コントロールを認識できない理由	396
Java AWT/Swing のサポート	396
Java AWT/Swing アプリケーションの属性	397
Java メソッドの動的な呼び出し	397
Java AWT/Swing テクノロジ ドメインでの priorLabel の判別	398
Oracle Forms のサポート	398
Java SWT と Eclipse RCP のサポート	399
Java SWT クラス リファレンス	400
Java SWT カスタム属性	400
Java SWT アプリケーションの属性	400
Java メソッドの動的な呼び出し	401
Java SWT と Eclipse アプリケーションのトラブルシューティング	402
モバイル アプリケーションのテスト	402
Android	403
iOS	409
モバイル アプリケーションの記録	422
テストを再生するモバイル デバイスの選択	422
Silk Test Workbench から UFT Mobile デバイスの直接使用	423
Silk Central 経由での UFT Mobile デバイスの使用	424
UFT Mobile との HTTPS 接続用証明書インストール	425
UFT Mobile パスワードの変更	426
Sauce Labs デバイスの使用	426
モバイル デバイスの接続文字列	426
モバイル デバイスの操作	430
モバイル デバイスの開放	430

モバイル アプリケーションのテストにおける setLocation メソッドの使用	432
Silk Central によって管理されているデバイスの使用	432
モバイル アプリケーションのテスト時のトラブルシューティング	432
モバイル Web アプリケーションのテストにおける制限事項	439
ネイティブ モバイル アプリケーションのテストにおける制限事項	440
ネイティブ モバイル アプリに対するメソッドの動的呼び出し	442
モバイル Web サイトでのオブジェクトのクリック	443
既存のモバイル Web テストの使用方法	444
.NET のサポート	444
Windows Forms のサポート	444
Windows Presentation Foundation (WPF) のサポート	449
Silverlight アプリケーションのサポート	458
Visual COBOL のサポート	462
Rumba のサポート	463
Rumba クラス リファレンス	463
Rumba の有効化と無効化	463
Rumba コントロールを識別するためのロケータ属性	464
Rumba での画面検証の使用	464
Unix ディスプレイのテスト	465
SAP のサポート	465
SAP クラス リファレンス	465
SAP アプリケーションの属性	465
SAP メソッドの動的な呼び出し	466
SAP コントロールの動的呼び出し	467
SAP の自動セキュリティ設定の構成	468
SAP eCATT の操作	468
ユニバーサル Windows プラットフォームのサポート	479
UWP アプリのテスト時のトラブルシューティング	480
Windows API ベースのアプリケーションのサポート	480
Win32 クラス リファレンス	481
Windows API ベースのクライアント/サーバー アプリケーションの属性	481
Win32 テクノロジ ドメインにおける priorLabel の決定方法	481
埋め込み Chrome アプリケーションのテスト	481
MFC (Microsoft Foundation Class) のサポート	482
クロス ブラウザ テスト	483
テストを再生するブラウザーの選択	484
xBrowser でのテスト オブジェクト	486
xBrowser オブジェクトに対するオブジェクト解決	487
xBrowser のページ同期	488
xBrowser における API 再生とネイティブ再生の比較	489
マウス移動の詳細設定	490
xBrowser のブラウザ構成の設定	490
リモート デスクトップ ブラウザーの接続文字列	492
リモート Windows マシン上でのブラウザーのテスト	492
Mac 上の Google Chrome または Mozilla Firefox のテスト	493
WebDriver ベースのブラウザーのケイパビリティの設定	493
Mac 上の Apple Safari を使用したテスト	495
Google Chrome を使用したテスト	499
Mozilla Firefox を使用したテスト	503
Microsoft Edge を使用したテスト	507
レスポンス Web デザインのテスト	509
ビジュアル ブレークポイントの検出	510
iframe パフォーマンスの向上	511
追加のブラウザーのバージョンでのテスト	512
クロス ブラウザ テスト:よくある質問	513

スクリプトからのブラウザの起動	518
非表示入力フィールドの検索	518
Web アプリケーションの属性	518
Web アプリケーションのカスタム属性	519
xBrowser クラス リファレンス	520
Microsoft Windows 8 および Microsoft Windows 8.1 上のテストの制限事項	520
サポートする属性の種類	520
Apache Flex アプリケーションの属性	520
Java AWT/Swing アプリケーションの属性	521
Java SWT アプリケーションの属性	521
SAP アプリケーションの属性	521
Silverlight コントロールを識別するためのロケータ属性	522
UI オートメーションを使ってコントロールを識別するためのロケータ属性	523
Rumba コントロールを識別するためのロケータ属性	524
Web アプリケーションの属性	524
Windows Forms アプリケーションの属性	525
Windows Presentation Foundation (WPF) アプリケーションの属性	525
Windows API ベースのクライアント/サーバー アプリケーションの属性	526
動的ロケータ属性	527
キーワード駆動テスト	528
キーワード駆動テストの利点	528
キーワード	529
Silk Test Workbench でキーワード駆動テストを作成する	530
Silk Test Workbench でのキーワード駆動テストの記録	530
Silk Test Workbench でのキーワード駆動テストの基本状態の設定	532
Silk Test Workbench でのキーワードの実装	532
Silk Test Workbench でのキーワードの記録	533
ビジュアル テストをキーワードとして指定	533
スクリプトのテスト メソッドをキーワードとして指定	534
キーワード間でのデータの受け渡し	534
キーワード駆動テストの編集	535
Silk Central でテストのキーワードを管理する	535
Silk Test Workbench のキーワード レコメンド機能	538
キーワードでのパラメータの使用	538
例：パラメータを取るキーワード	539
キーワードのキーワード シーケンスへの結合	541
変数を指定したキーワード駆動テストの再生	541
Silk Test Workbench と Silk Central の統合	542
Silk Central からキーワードを開く	542
Silk Test Workbench での Silk Central キーワードの実装	543
Silk Central へのキーワード ライブラリのアップロード	544
コマンド ラインから Silk Central へのキーワード ライブラリの更新	544
キーワードの検索	546
キーワードのフィルタリング	546
キーワードのグループ化	546
キーワード駆動テストのトラブルシューティング	547
コマンドライン	548
SilkTest.Exe コマンド ライン	548
STW.EXE コマンド ライン	548
STW.EXE のパラメータ	548
入力ファイル	553
出力ファイル	554
エラー	555
Silk Central から STW.EXE を実行するための前提条件	558
STW.EXE の使用例	558

コマンドラインからの資産のインポートとエクスポート	560
資産のインポートのコマンドラインパラメータ	561
資産のエクスポートのコマンドラインパラメータ	564
ログ	566
Silk Test Workbench のオプション	568
グローバルオプション	568
eCATT	568
全般オプション	569
全般オプションの変更	570
開始画面オプション	571
開始画面オプションの変更	572
記録オプション	572
イメージ資産記録設定の設定	573
オブジェクトマップの詳細設定の設定	573
マウス移動の詳細設定	574
無視するクラスの設定	575
ロケータの詳細設定の設定	575
UIオートメーションオプションの設定	576
ブラウザの記録オプションの設定	577
カスタム属性の設定	578
記録/再生の対象とする WPF クラスの設定	579
記録の出力オプションの設定	579
記録のホットキーオプションの設定	582
記録オプションプロファイルの作成	582
保存済み記録オプションの適用	583
再生オプション	583
全般再生オプションの設定	584
xBrowser同期オプションの設定	585
再生のタイミングオプションの設定	586
UIオートメーションオプションの設定	587
閉じるオプションの設定	588
再生結果オプションの設定	588
再生ホットキーの変更	591
再生オプションプロファイルの作成	591
保存済み再生オプションの適用	591
スクリプト作成オプション	592
詳細オプションの設定	592
アクティブデータ詳細設定オプション	593
Silk Test Workbench の管理	595
Silk Test Workbench の実行に必要な管理者権限	595
Silk Test Workbench データベースの構成	595
SQL Server データベースの構成	595
Oracle データベースの構成	598
ドメインなしでの SQL Server または Oracle データベースの設定	604
データソース名の作成	605
データベースへ接続する	607
データベース構成の共有	609
データベースメンテナンス	610
Silk Test Workbench データベースを開く	610
データベースバージョンの更新	612
データベースレコードのコピー	612
データベースのレコードのロック解除	613
Access データベースのコンパクト化	613
データベースメンテナンス設定の構成	614
接続ユーザーがいるデータベースのメンテナンスの実行	614

データベース整合性チェックの実行	615
データベースの膨張の抑制	615
プロジェクトの管理	615
プロジェクトの追加	616
プロジェクトの参照	616
プロジェクト参照の表示	617
グローバルプロジェクトの定義	617
Silk Test Workbench がプロジェクトの循環参照を回避する方法	617
プロジェクト名の変更	619
プロジェクトの複製	620
プロジェクトの削除	620
ユーザーの管理	621
データベースに接続中のユーザーの表示	621
ユーザーの追加	621
ユーザーの削除	622
ユーザー プロファイルの編集	622
プロジェクトに対するユーザーのアクセス権の変更	622
データベースからのユーザーのログアウト	623
グループの管理	623
グループの追加	623
グループの削除	624
グループの変更	624
UI トピック	625
開始画面	625
Silk Test Workbench のメニュー	626
[ファイル] メニュー	626
[編集] メニュー	627
[操作] メニュー	628
[表示] メニュー	628
[挿入] メニュー	630
[デバッグ] メニュー	631
[ツール] メニュー	632
[ウィンドウ] メニュー	634
[ヘルプ] メニュー	634
ツールバー	634
ビジュアルナビゲータ	637
[ビジュアルナビゲータ]: [テストステップ] ペイン	639
[ビジュアルナビゲータ]: [画面プレビュー]	643
[ビジュアルナビゲータ]: [プロパティ]	644
[ビジュアルナビゲータ]: [ストーリーボード]	645
ビジュアルナビゲータのレイアウトのカスタマイズ	646
[結果] ウィンドウ	646
アセットブラウザ	647
[オブジェクトの識別] ダイアログ ボックス	648
[コード] ウィンドウ	650
スクリプトの [プロパティ] ペインの表示	650
出カウィンドウ	651
既知の問題	652
全般的な問題	652
モバイル Web アプリケーション	654
Web アプリケーション	654
Google Chrome	654
Internet Explorer	656
Microsoft Edge	657
Mozilla Firefox	658

SAP アプリケーション	658
Silk Test Workbench	659
Silk Test Workbench 言語リファレンス	663
Common Class Reference	663
BaseGuiTestObject Class	663
CharSet Enumeration	667
CheckBox Class	667
CheckBoxToolItem Class	672
ComboBox Class	675
Control Class	680
SystemFunctions Class	684
Desktop Class	687
Dialog Class	687
DropDownToolItem Class	691
ExecutionMode Enumeration	695
ExecutionResult Class	696
FileHandle Class	696
FileInfo Class	696
FileOpenMode Enumeration	697
FilePointerMode Enumeration	697
FileShareMode Enumeration	698
FileSizeUnit Enumeration	698
Group Class	699
GuiTestObject Class	703
HorizontalScrollBar Class	706
IClickable Interface	711
IFocusable Interface	711
IKeyable Interface	711
IMoveable Interface	712
INativeWindow Interface	713
IScrollable Interface	713
Item Class	713
Label Class	717
Link Class	721
ListBox Class	725
Menu Class	729
MenuItem Class	733
PushButton Class	737
PushToolItem Class	741
RadioList Class	745
RadioListToolItem Class	749
RegistryCategory Enumeration	753
RegistryView Enumeration	753
Scale Class	754
ScrollBar Class	758
SeparatorItem Class	762
TabControl Class	766
Table Class	770
TableColumn Class	774
TableRow Class	778
TestObject Class	782
TextField Class	784
ToggleButton Class	789
ToolBar Class	793
ToolItem Class	797

Tree Class	800
VerticalScrollBar Class	805
Window Class	809
コア クラス リファレンス	814
ActiveData クラス	814
ActiveDataRow クラス	819
Agent クラス	825
BaseState クラス	831
BrowserBaseState クラス	834
ConsoleWindow クラス	835
DllCall クラス	839
IBaseState インターフェイス	840
Timer クラス	842
Workbench クラス	845
Data Types	853
ClickType Enumeration	853
Color Class	853
ItemIdentifier Data Type	854
ItemPath Data Type	855
ModifierKeys Enumeration	855
MouseButton Enumeration	856
Point Class	856
Range Class	857
Rectangle Class	857
TextPosition Class	858
TextRange Class	858
TreeContent Class	859
TreeNode Class	859
Flex Class Reference	859
FlexAccordion Class	860
FlexAdvancedDataGrid Class	867
FlexAlert Class	876
FlexApplication Class	883
FlexAreaChart Class	889
FlexAreaSeries Class	896
FlexAxisRenderer Class	902
FlexBarChart Class	908
FlexBarSeries Class	916
FlexBox Class	922
FlexBubbleChart Class	928
FlexBubbleSeries Class	936
FlexButton Class	942
FlexButtonBar Class	948
FlexCandlestickChart Class	955
FlexCandlestickSeries Class	962
FlexCanvas Class	968
FlexCartesianChart Class	975
FlexChart Class	982
FlexChartLegend Class	988
FlexChartSeries Class	994
FlexCheckBox Class	1000
FlexColorPicker Class	1006
FlexColumnChart Class	1013
FlexColumnSeries Class	1020
FlexComboBase Class	1026

FlexComboBox Class	1033
FlexContainer Class	1040
FlexContainerMovieClip Class	1046
FlexDataGrid Class	1052
FlexDateChooser Class	1060
FlexDateField Class	1066
FlexDisplayObject Class	1074
FlexDividedBox Class	1078
FlexForm Class	1085
FlexFormItem Class	1091
FlexHLOCChart Class	1098
FlexHLOCSeries Class	1105
FlexHLOCSeriesBase Class	1111
FlexImage Class	1117
FlexLabel Class	1123
FlexLineChart Class	1128
FlexLineSeries Class	1135
FlexLinkBar Class	1142
FlexList Class	1148
FlexListBase Class	1156
FlexListLabel Class	1163
FlexLoader Class	1168
FlexMenu Class	1173
FlexMenuBar Class	1179
FlexNavigationBar Class	1185
FlexNumericStepper Class	1192
FlexObject Class	1198
FlexOLAPDataGrid Class	1203
FlexPanel Class	1213
FlexPieChart Class	1220
FlexPieSeries Class	1226
FlexPlotChart Class	1233
FlexPlotSeries Class	1240
FlexPopUpButton Class	1246
FlexProgressBar Class	1252
FlexRadioButton Class	1258
FlexRepeater Class	1264
FlexRule Class	1268
FlexScrollBar Class	1273
FlexScrollBase Class	1279
FlexSlider Class	1284
FlexStandalonePlayer Class	1290
FlexTabNavigator Class	1295
FlexTextArea Class	1302
FlexTitleWindow Class	1308
FlexToggleButtonBar Class	1315
FlexTree Class	1322
FlexUIMovieClip Class	1330
FlexVideoDisplay Class	1335
FlexViewStack Class	1340
FlexWindow Class	1347
FlexWindowedApplication Class	1354
SparkAirHTML Class	1362
SparkApplication Class	1367
SparkBorderContainer Class	1373

SparkButton Class	1379
SparkButtonBar Class	1385
SparkButtonBarButton Class	1391
SparkButtonBase Class	1397
SparkCheckBox Class	1402
SparkComboBox Class	1408
SparkComplexDisplay Class	1415
SparkDataGrid Class	1422
SparkDataGridLabel Class	1428
SparkDataGroup Class	1433
SparkDataRenderer Class	1440
SparkDropDownList Class	1446
SparkDropDownListBase Class	1453
SparkForm Class	1460
SparkFormItem Class	1466
SparkGroup Class	1472
SparkGroupBase Class	1479
SparkImage Class	1485
SparkLabel Class	1491
SparkList Class	1496
SparkListBase Class	1503
SparkListLabel Class	1509
SparkMuteButton Class	1516
SparkNavigatorContent Class	1521
SparkNumericStepper Class	1527
SparkObject Class	1533
SparkPanel Class	1538
SparkPopUpAnchor Class	1544
SparkRadioButton Class	1550
SparkRange Class	1556
SparkRichEditableText Class	1561
SparkRichText Class	1567
SparkScrollBar Class	1573
SparkSkinnableContainer Class	1579
SparkSkinnableContainerBase Class	1585
SparkSkinnableDataContainer Class	1590
SparkSkinnablePopUpContainer Class	1596
SparkSkinnableTextBase Class	1602
SparkSlider Class	1608
SparkSpinner Class	1614
SparkTabBar Class	1619
SparkTextArea Class	1626
SparkTextBase Class	1632
SparkTextInput Class	1638
SparkTileGroup Class	1644
SparkTitleWindow Class	1651
SparkToggleButton Class	1657
SparkToggleButtonBase Class	1663
SparkTrackBase Class	1668
SparkVideoDisplay Class	1674
SparkVideoPlayer Class	1679
SparkVolumeBar Class	1685
SparkWindow Class	1690
SparkWindowedApplication Class	1697
Java SWT Class Reference	1705

CBanner Class	1705
CoolBar Class	1709
CoolItem Class	1713
CTabFolder Class	1716
CTabItem Class	1721
ExpandBar Class	1725
ExpandItem Class	1729
HorizontalSash Class	1733
Sash Class	1737
SashForm Class	1741
ScrollableControl Class	1745
ScrolledComposite Class	1749
Shell Class	1753
Spinner Class	1757
StyledText Class	1761
SWTBrowser Class	1766
SWTDateTime Class	1770
SWTTabControl Class	1774
SWTTabItem Class	1778
SWTTable Class	1782
SWTTableColumn Class	1786
SWTTableRow Class	1790
SWTTree Class	1794
SWTTreeColumn Class	1799
VerticalSash Class	1803
ViewForm Class	1807
キーワード駆動テストクラス リファレンス	1811
ArgumentAttribute クラス	1811
KeywordAttribute クラス	1812
KeywordGroupAttribute クラス	1813
Mobile Class Reference	1813
IMobileClickable Interface	1813
IMobileGestures Interface	1814
IMobileKeyable Interface	1814
MobileButton Class	1815
MobileDevice Class	1819
MobileObject Class	1823
MobileTextField Class	1827
MobileWindow Class	1831
Rumba Class Reference	1835
Rumba Keys	1835
RumbaCharacterAttribute Data Type	1838
RumbaField Class	1839
RumbaLabel Class	1843
RumbaObject Class	1847
RumbaScreen Class	1851
RumbaTextField Class	1855
SAP Class Reference	1860
ISapContextMenuable Interface	1860
SapBarChart Class	1860
SapBox Class	1865
SapButton Class	1869
SapCalendar Class	1873
SapChart Class	1877
SapCheckBox Class	1881

SapColorSelector Class	1885
SapComboBox Class	1890
SapComponent Class	1894
SapContainer Class	1898
SapContainerShell Class	1902
SapContextMenu Class	1906
SapCustomControl Class	1909
SapDockShell Class	1913
SapGridView Class	1917
SapHorizontalScrollBar Class	1924
SapHTMLViewer Class	1927
SapLabel Class	1932
SapMenu Class	1936
SapMenubar Class	1940
SapNetPlan Class	1943
SapOfficeIntegration Class	1948
SapOkCodeField Class	1952
SapPicture Class	1956
SapRadioButton Class	1960
SapScrollbar Class	1964
SapScrollContainer Class	1967
SapShell Class	1971
SapSimpleContainer Class	1975
SapSplitterContainer Class	1979
SapStatusbar Class	1983
SapTab Class	1987
SapTable Class	1991
SapTabStrip Class	1995
SapTextEdit Class	1999
SapTextField Class	2004
SapTitlebar Class	2008
SapToolbar Class	2012
SapToolbarControl Class	2016
SapTree Class	2020
SapUserArea Class	2027
SapVerticalScrollBar Class	2031
SapWindow Class	2034
Silverlight Class Reference	2039
SLApplication Class	2039
SLAutoCompleteBox Class	2043
SLBase Class	2048
SLButton Class	2053
SLCalendar Class	2056
SLCalendarButton Class	2061
SLCalendarDayButton Class	2065
SLCheckBox Class	2068
SLComboBox Class	2072
SLComboBoxItem Class	2077
SLDataGrid Class	2081
SLDataGridCell Class	2086
SLDataGridDetails Class	2089
SLDataGridRow Class	2093
SLDataPager Class	2097
SLDatePicker Class	2101
SLDescriptionViewer Class	2105

SLFrame Class	2109
SLGridSplitter Class	2113
SLGroup Class	2117
SLHeader Class	2120
SLHeaderItem Class	2124
SLHorizontalScrollBar Class	2128
SLHyperlinkButton Class	2132
SLImage Class	2136
SLListBox Class	2140
SLListItem Class	2144
SLMediaElement Class	2148
SLMenu Class	2152
SLMenuBar Class	2156
SLMenuItem Class	2160
SLMultiScaleImage Class	2163
SLPane Class	2167
SLPasswordBox Class	2171
SLPopup Class	2175
SLProgressBar Class	2179
SLRadioButton Class	2183
SLRepeatButton Class	2187
SLRichTextBox Class	2190
SLSeparator Class	2194
SLSlider Class	2198
SLSpinner Class	2202
SLSplitButton Class	2206
SLStatusBar Class	2210
SLTabControl Class	2214
SLTabItem Class	2218
SLTable Class	2222
SLTextBlock Class	2225
SLTextBox Class	2229
SLThumb Class	2233
SLTitleBar Class	2237
SLToggleButton Class	2241
SLToolBar Class	2245
SLToolTip Class	2248
SLTreeView Class	2252
SLTreeViewItem Class	2257
SLValidationSummary Class	2261
SLVerticalScrollBar Class	2265
SLWindow Class	2269
Java AWT and Swing Class Reference	2272
AbstractButton Class	2273
Applet Class	2277
AppletContainer Class	2281
AWTButton Class	2285
AWTCanvas Class	2289
AWTCheckbox Class	2293
AWTCheckboxMenuItem Class	2297
AWTChoice Class	2301
AWTComponent Class	2305
AWTContainer Class	2309
AWTDialog Class	2314
AWTFrame Class	2318

AWTHorizontalScrollbar Class	2323
AWTLabel Class	2327
AWTList Class	2332
AWTMenu Class	2336
AWTMenuComponent Class	2340
AWTMenuItem Class	2343
AWTRadioButton Class	2347
AWTScrollbar Class	2351
AWTScrollPane Class	2355
AWTTextArea Class	2360
AWTTextComponent Class	2364
AWTTextField Class	2369
AWTVerticalScrollbar Class	2374
AWTWindow Class	2378
BasicArrowButton Class	2383
IAWTScrollable Interface	2387
IAWTScroller Interface	2388
IBaseScrollable Interface	2388
IOracleFormsMenuBase Interface	2389
IOracleFormsScrollable Interface	2389
IOracleFormsScroller Interface	2390
JButton Class	2391
JCheckBox Class	2395
JCheckBoxMenuItem Class	2399
JColorChooser Class	2404
JComboBox Class	2408
JComponent Class	2412
JDesktopPane Class	2417
JDialog Class	2421
JEditorPane Class	2425
JFrame Class	2430
JHorizontalScrollBar Class	2435
JLabel Class	2439
JLayeredPane Class	2443
JList Class	2447
JMenu Class	2452
JMenuBar Class	2456
JMenuItem Class	2460
JPanel Class	2465
JPasswordField Class	2469
JPopupMenu Class	2473
JProgressBar Class	2477
JRadioButton Class	2482
JRadioButtonMenuItem Class	2486
JRootPane Class	2490
JScrollBar Class	2494
JScrollPane Class	2499
JSlider Class	2503
JSpinner Class	2507
JSplitPane Class	2512
JTabbedPane Class	2516
JTable Class	2520
JTableHeader Class	2525
JTextArea Class	2529
JTextComponent Class	2534

JTextField Class	2539
JTextPane Class	2543
JToggleButton Class	2548
JToolBar Class	2552
JTree Class	2557
JVerticalScrollBar Class	2561
JViewport Class	2566
JWindow Class	2570
OracleFormsApplication Class	2574
OracleFormsButton Class	2579
OracleFormsCheckbox Class	2583
OracleFormsChoice Class	2587
OracleFormsComboBox Class	2591
OracleFormsContainer Class	2596
OracleFormsHorizontalScrollbar Class	2600
OracleFormsLabel Class	2605
OracleFormsListBox Class	2609
OracleFormsListView Class	2614
OracleFormsMenu Class	2620
OracleFormsMenuItem Class	2624
OracleFormsPopList Class	2628
OracleFormsPopupMenu Class	2633
OracleFormsRadioButton Class	2637
OracleFormsScrollbar Class	2641
OracleFormsStatusArea Class	2646
OracleFormsStatusBar Class	2650
OracleFormsStatusBarItem Class	2654
OracleFormsStatusIndicator Class	2658
OracleFormsTabBar Class	2662
OracleFormsTabBarItem Class	2666
OracleFormsTabPanel Class	2670
OracleFormsTextField Class	2674
OracleFormsTitleBar Class	2679
OracleFormsToolBar Class	2683
OracleFormsToolBarItem Class	2687
OracleFormsTree Class	2692
OracleFormsVerticalScrollbar Class	2697
SplitPaneDivider Class	2702
Win32 Class Reference	2706
AccessibleControl Class	2706
Header Class	2710
ListView Class	2714
MonthCalendar Class	2718
Pager Class	2723
ProgressBar Class	2727
StatusBar Class	2731
UpDown Class	2734
Windows Forms Class Reference	2739
CheckedListBox Class	2739
DataGrid Class	2743
DataGridColumn Class	2747
DataGridItem Class	2751
DataGridRow Class	2755
DomainUpDown Class	2759
ElementHost Class	2763

FormsHost Class	2767
FormsWindow Class	2771
MenuStrip Class	2776
NumericUpDown Class	2780
xBrowser Class Reference	2784
BrowserApplication Class	2784
BrowserObject Class	2790
BrowserWindow Class	2793
DomButton Class	2797
DomCheckBox Class	2802
DomDataList Class	2806
DomElement Class	2811
DomEmbeddedElement Class	2816
DomForm Class	2820
DomLink Class	2825
DomListBox Class	2829
DomRadioButton Class	2834
DomTable Class	2838
DomTableRow Class	2843
DomTextField Class	2847
IItemSelectable Interface	2852

Silk Test Workbench 21.0 へようこそ

発行日 :



Silk Test Workbench 21.0 へようこそ

はじめに
ドキュメント
製品スイート



新機能

Silk Test Workbench 21.0 の新機能



主なセクション

テストの作成
テストの再生と結果の分析
テストのデバッグ
オブジェクト解決



チュートリアルとデモ

ビジュアルテスト チュートリアル
スクリプト チュートリアル



コード サンプル

言語リファレンス



オンライン リソース

Micro Focus Web サイト
Silk Test YouTube ビデオ
Silk Test オンライン ドキュメント
Silk Test On-Demand Training
Silk Test サポートとサービス
Silk Test 製品アップデート
Silk Test Community
Silk Test Blog
Silk Test Knowledge Base



フィードバックをお寄せください

[Micro Focus へのお問い合わせ \(76 ページ\)](#)

[このヘルプに関するフィードバックを電子メールで送る](#)



ライセンス情報

評価版を使用しているのではない限り、Silk Test はライセンスを必要とします。



注: Silk Test ライセンスは、Silk Test の特定のバージョンに固定されています。たとえば、Silk Test 21.0 には Silk Test 21.0 のライセンスが必要です。

ライセンス モデルは、使用しているクライアントとテストすることができるアプリケーションに基づきま
す。利用可能なライセンス モードに応じて、次のアプリケーションの種類がサポートされます。

ライセンス モード	アプリケーションの種類
モバイル ネイティブ	<ul style="list-style-type: none">モバイル Web アプリケーション<ul style="list-style-type: none">AndroidiOSネイティブ モバイル アプリケーション<ul style="list-style-type: none">AndroidiOS
フル	<ul style="list-style-type: none">Web アプリケーション (以下を含む)<ul style="list-style-type: none">Apache FlexJava アプレットモバイル Web アプリケーション<ul style="list-style-type: none">AndroidiOSApache FlexJava AWT/Swing (Oracle Forms を含む)Java SWT と Eclipse RCP.NET (Windows Forms および Windows Presentation Foundation (WPF) を含む)RumbaWindows API ベース <p> 注: ライセンスをフル ライセンスにアップグレードする場合は、http://www.microfocus.co.jp に移動します。</p>
プレミアム	フル ライセンスでサポートされるすべてのアプリケーションの種類 + SAP アプリケーション <p> 注: ライセンスをプレミアム ライセンスにアップグレードする場合は、http://www.microfocus.co.jp に移動します。</p>
モバイル ネイティブ アドオン	フル ライセンスとプレミアム ライセンスでサポートされるテクノロジーに加えて、モバイル ネイティブ アドオン ライセンスによって、Android と iOS 上でのネイティブ モバイル アプリケーション テストのサポートを提供します。

はじめに

Silk Test Workbench は、自分たちのテストでの使いやすさを重視するビジネス アナリスト、ドメイン エキスパート、オートメーション エキスパートにとって理想的な Silk Test クライアントです。Silk Test Workbench にはビジュアル テスト オプションがあり、コードを作成せずにテストを視覚的に作成できる、直感的にわかりやすい強力な方法を提供します。スクリプト作成オプションでは、さらに強力で柔軟性の高いテストを実行でき、ビジュアル テストと組み合わせて VB.NET スクリプトを拡張できます。Silk Test Workbench を使用すると、アプリケーションとのユーザー セッションを記録してテストを作成し、検証やテスト ロジックを追加してテストを拡張し、テストを再生してアプリケーションが期待どおりに動作することを確認することができます。

このセクションでは、Silk Test Workbench の概要と Silk Test Workbench インターフェイスの概要について説明します。

ログオン

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > クライアント > Silk Test Workbench**、または (Microsoft Windows 10) **スタート > Silk > Silk Test Workbench** をクリックします。Silk Test Workbench が起動され、**SilkTest Workbench にログイン** ダイアログ ボックスが表示されます。

2. **認証** リストボックスから、**Silk Test Workbench** 認証、または **Windows** 認証のいずれかを使用する方を選択します。



注: Windows 認証を使用できるユーザーは、**ツール > 管理** を使用して Silk Test Workbench ユーザーに追加されたユーザーだけです。

3. **Silk Test Workbench** 認証を使用している場合、**ユーザー名** フィールドにユーザー名を入力し、**パスワード** フィールドにパスワードを入力します。**Windows** 認証を使用している場合は、ユーザー名とパスワードを入力する必要はありません。



注: Silk Test Workbench をはじめて使用する場合は、管理者権限を持つユーザー名とパスワードが必要になる場合があります。デフォルトのユーザー名は Admin で、デフォルトのパスワードは admin です。不正アクセスを防ぐため、このパスワードはログオンのあとで変更してください。ログオンパスワードはいつでも変更できます。

4. **データベース** リストから、使用するデータベースを選択します。

データベースが **データベース** リストに表示されるようにするには、まずデータベース接続を使用できるように構成する必要があります。データベース接続を構成するには、**ログオン** ダイアログ ボックスで **構成** ボタンをクリックします。

5. **OK** をクリックします。

Silk Test Workbench データベースに初めてログインすると、*Project1* という名前のデフォルトプロジェクトがデータベースに作成されます。デフォルトでは、作成する資産は、このプロジェクトに保存されます。プロジェクト名は、**プロジェクトの編集** をクリックすると変更できます。

開始画面 が開きます。Silk Test Workbench に始めてログインすると、Silk Test Workbench は *Project1* または最初に利用可能な非グローバル オブジェクトをアクティブ オブジェクトとして選択します。

ログオン パスワードの変更

ログオン パスワードを変更するには、以下の手順に従います。

1. **ツール > パスワードの変更** をクリックします。**パスワードの変更** ダイアログ ボックスが開きます。現在のユーザーのログオン名が、ダイアログ ボックスのタイトル バーに表示されます。

2. パスワードを変更し、**OK** をクリックします。このユーザー ID を使って次回ログオンしたとき、変更が有効になります。

Silk Test Workbench の概要

Silk Test Workbench は、Microsoft、Java、Web、その他多くの分散テクノロジーで開発された複雑なアプリケーションの機能テストを高速化する自動テストツールです。Silk Test Workbench を使用すると、アプリケーションとのユーザー セッションを記録してテストを作成し、検証やテスト ロジックを追加してテストを拡張し、テストを再生してアプリケーションが期待どおりに動作することを確認することができます。

Silk Test Workbench には、テストの自動化を開発するための 2 つのオプションがあります。以下を使用できます。

- ビジュアル テスト
- VB ベースの .NET スクリプト

Silk Test Workbench によりユーザー操作を確実に記録し、強力なビジュアル テストをすばやく作成できます。記録したビジュアル テストはそれぞれ、わかりやすい簡潔なステップの一連の操作として表示されます。これは初心者からエキスパートまであらゆるテスト担当者が容易に理解できるものです。テスト担当者は、注文の処理や顧客情報の入力など、すべてのユーザー操作について Silk Test Workbench の記録方法を広範に制御できます。テスト担当者は、記録中または記録終了後いつでもテストを編集して、テストプロジェクトの最も困難な要求にも合わせるすることができます。

Silk Test Workbench は、さまざまな開発ツールで開発されたアプリケーションのテストをサポートします。テスト ロジック、実際のデータ、検証、およびエラー処理をテストに追加して、さらに柔軟性と信頼性を上げることが可能です。

Silk Test Workbench は、テスト、開発、品質保証、その他のプロジェクト関係者間でのテスト ステータスの効率的な伝達をサポートするプロジェクト協業機能により、チームの生産性を向上させます。アドホッククエリと詳細なレポートは、チームがテストの実行結果を解釈して、情報に基づきテストプロジェクトに関する意思決定を行うことを可能にします。

Silk Test Workbench を使用する利点

Silk Test Workbench は、アプリケーション ソフトウェア、Web アプリケーション、およびモバイル アプリケーションのテスト手順の計画、開発、実行とバグの検出を支援します。Silk Test Workbench では、アプリケーションのセッションを記録し、実際の結果が予想した結果と一致するかどうかを調べる検証機能を追加したうえで、いつでもセッションを再生してアプリケーションが期待どおりに機能するかを確認できます。

Silk Test Workbench は、開発チームに以下のような利点を提供します。

- | | |
|----------------------|---|
| 開発/テストサイクルの短縮 | Silk Test Workbench では、マシンの使用時間をスケジュールしやすい夜間や週末に無人でテストを再生できるため、開発とテストのスケジュールを早めてアプリケーションをより早く市場に出すことができます。 |
| 一貫性 | Silk Test の自動テストは毎回同じように実行されるため、再現可能な一貫した結果が生成されます。 |
| 生産性の向上 | Silk Test Workbench は手動操作なしでテストを再生できるため、プログラマーやテスト担当者は他の作業に時間を使うことができます。 |
| 迅速なテスト開発 | アプリケーションをナビゲートしてテストしながらユーザーの操作を記録することにより、有用なテストをすばやく簡単に作成できます。テンプレートを設定して、テストの標準フレームワークを作成することもできます。自動テストは、手動によるテストの数分の 1 の時間で実行できます。 |

テスト結果の確認 テストスクリプトとビジュアル テスト再生の詳細は結果に記録され、いつでも確認できます。

特殊な状況下での自動化 (周辺機器が無い)

製品の基本的な位置付け

Silk Test Workbench は GUI テスト製品で、自動化された状況下での有意なテスト結果を得るために、人間のように振舞います。Silk Test Workbench が実行したテストは、人間が実行するよりもすばやく実行しますが、同等の価値のあるものです。このことは、人間が同じテストを実行するために必要なテスト環境とできる限り同等なテスト環境を Silk Test Workbench が必要とすることを意味します。

物理的な周辺機器

実際のアプリケーション UI の手動テストでは、キーボード、マウス、ディスプレイなどの入出力デバイスが必要です。Silk Test Workbench では、テストの再生時に物理的な入力デバイスを必要としません。Silk Test Workbench に必要なものは、キーストロークやマウス クリックを実行するオペレーティング システムの機能です。大抵の場合、入力デバイスが接続されていなくても、Silk Test Workbench の再生は期待通り動作します。ただし、デバイス ドライバによっては、物理的な入力デバイスが利用可能でないと、Silk Test Workbench の再生機構をブロックする場合があります。

同じことが物理的な出力デバイスについても言えます。物理的なディスプレイが接続されている必要はありませんが、機能するビデオ デバイス ドライバがインストールされ、オペレーティング システムが画面にレンダリングできる状態になければなりません。たとえば、スクリーン セーバー モードやセッションがロックされている状況では、レンダリングできません。レンダリングできない場合、低レベルの再生は機能せず、高レベルの再生もテスト対象アプリケーション (AUT) で使用するテクノロジーに依存しますが、期待通り機能しない可能性があります。

仮想マシン

Silk Test Workbench は仮想化ベンダーを直接サポートしませんが、仮想ゲスト マシンが物理マシンと同等に動作する限り、任意の仮想化手法のもとで動作可能です。標準的な周辺機器は、通常は仮想デバイスとして提供されており、仮想マシンを実行するマシンで使用されている物理デバイスとは無関係です。

クラウド インスタンス

自動化の観点からは、クラウド インスタンスは仮想マシンと変わりありません。ただし、クラウド インスタンスでは、ビデオ レンダリングに特殊な最適化が行われている場合があります。ハードウェア リソースの消費を抑えるために、画面のレンダリングが一時的にオフになる状況があります。これは、ディスプレイを表示しているアクティブなクライアントが無いと、クラウド インスタンスが検知した場合に発生する場合があります。このような場合、回避策として VNC ウィンドウを開くことができます。

特殊な状況

ウィンドウが無く起動されるアプリケーション (ヘッドレス) このようなアプリケーションは、Silk Test Workbench を使ってテストできません。Silk Test Workbench は、対象のアプリケーション プロセスにフックして、対話操作する必要があります。ウィンドウが表示されないプロセスをフックすることはできません。このような場合は、システム コマンドの実行のみ可能です。

リモート デスクトップ、ターミナル サービス、リモート アプリケーション (すべてのベンダー) Silk Test Workbench がリモート デスクトップ セッション側に存在し、操作する場合、完全に期待通りの操作が行われます。



注: フル ユーザー セッションが必要で、リモート表示ウィンドウは最大化されている必要があります。リモート表示ウィンドウが何らかの理由で表示されていない場合 (ネットワーク上の問題など)、Silk Test Workbench は再生を続けますが、使用されているリモート表示技術によっては予期しない結果を生じ

る可能性があります。たとえば、リモートデスクトップセッションが失われると、ビデオレンダリングに悪影響を与えますが、他のリモート表示手法では、一度表示されたウィンドウが失われても、問題なく表示されるものもあります。

Silk Test Workbench がリモートデスクトップ、リモートビュー、リモート アプリ ウィンドウなどとの対話操作に使用される場合は、Silk Test Workbench が見ることができるのはリモート マシンのスクリーンショットだけであるため、低レベルな技術だけが使用できます。リモート表示技術によっては、セキュリティ上の制約により、低レベル操作でさえできないものもあります。たとえば、リモート アプリケーション ウィンドウにキーストロークを送信できない場合があります。

既知の自動化の障壁

Silk Test Workbench では、ログオンした対話的なフル ユーザー セッションが必要です。スクリーンセーバー、休止状態、スリープ モードなどのセッションをロックするものは無効化してください。組織の方針などで、これができない場合は、キーボード アライブ 操作 (定期的にあるいは各テストケースの終わりにマウスを動かすなど) を追加することによって、このような問題を回避できます。



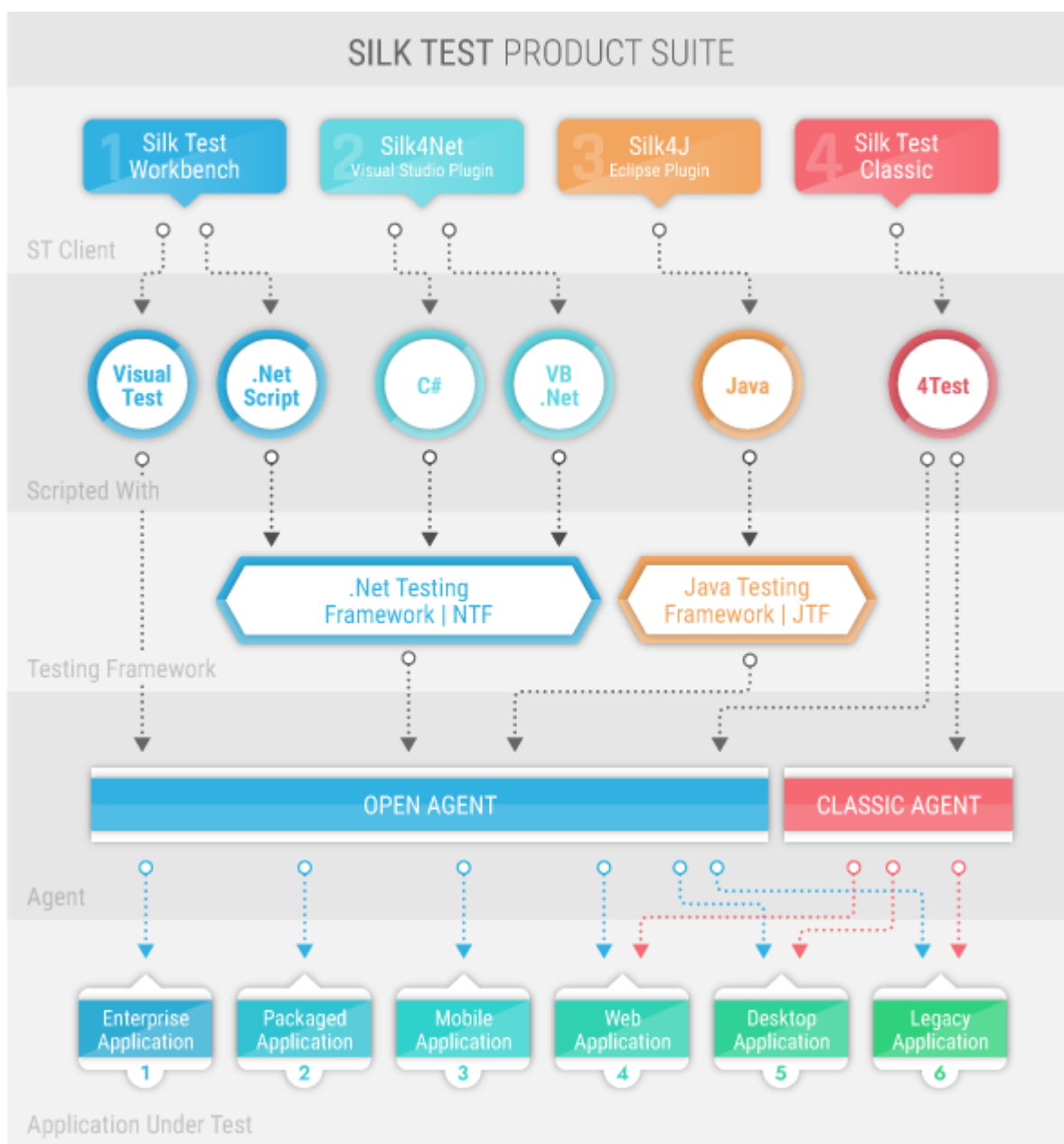
注: 実際のテスト環境の構成や AUT、仮想化、ターミナル サービスで使用される技術によっては、テストの自動化プロセスにおいて、さらなる問題や制約に直面する可能性があります。

Silk Test 製品スイート

Silk Test は、高速で信頼性の高い機能テストと回帰テストを行うための自動テスト ツールです。Silk Test は、高品質のソフトウェアをすばやくリリースするために、開発チーム、品質管理チーム、ビジネス アナリストを支援します。Silk Test を使用すると、アプリケーションが意図したとおりに動作することを確実にするために、複数のプラットフォームとデバイス上でテストを記録/再生することができます。

Silk Test 製品スイートには、以下のコンポーネントが含まれています。

- Silk Test Workbench : Silk Test Workbench は、品質テスト環境です。上級者用の .NET スクリプトと、より幅広い利用者がテストを行えるようにする使いやすいビジュアルテストが提供されます。
- Silk4NET : Silk4NET Visual Studio プラグインを使用すると、Visual Studio で直接 VB.NET または C# のテスト スクリプトを作成できます。
- Silk4J : Silk4J Eclipse プラグインを使用すると、Eclipse 環境で直接 Java ベースのテスト スクリプトを作成できます。
- Silk Test Classic : Silk Test Classic は、4Test を使用したスクリプトを作成できる Silk Test クライアントです。
- Silk Test Agent : Silk Test Agent は、テストのコマンドを GUI 固有のコマンドに変換するソフトウェア プロセスです。つまり、テストするアプリケーションをエージェントが動かし、監視しています。ホストマシン上で 1 つのエージェントをローカルに実行できます。ネットワーク環境では、任意の数のエージェントをリモート マシン上で実行できます。



上の画像の個々のボックスのサイズは、記述上の理由で異なっているだけで、含まれる機能を反映しているわけではありません。

インストールする製品スイートによって、使用できるコンポーネントが決まります。すべてのコンポーネントをインストールするには、完全インストール オプションを選択します。Silk Test Classic を除くすべてのコンポーネントをインストールするには、標準インストール オプションを選択します。

Silk Test Workbench の機能

Silk Test Workbench はテスト担当者の動作を模倣します。キー入力やマウス操作、メニューやリストからの項目の選択など、テスト担当者と同じようにアプリケーションを操作します。ただし、キー入力やマウス操作は、ハードウェアではなくソフトウェアによって自動的に生成されます。Silk Test Workbench は、アプリケーションが入力に対してどのように反応するか、画面を監視することによって、テスト操作を決定します。

予想結果と実際の結果が一致しないテスト条件にフラグを付けることができます。Silk Test Workbench のテストは柔軟です。スクリプトは、前の操作の結果に基づいて、実行すべき操作を決定できます。Silk Test Workbench は、システム クロックやハード ディスクなど、PC の内部リソースにもアクセスできるため、時間を考慮したり、ファイルの読み書きを行ったりもできます。

開始画面 と **ビジュアル ナビゲータ** は、テスト ソリューションを迅速に提供できるように設計されています。**開始画面** でテスト プロジェクトを管理し、**ビジュアル ナビゲータ** を使用して個々のテスト ステップを制御できます。

一元的マルチユーザー資産データベースは、テスト プロセスを管理するうえで効果的です。テスト実行に対するレポート作成機能も提供されます。

Silk Test Workbench は、動的オブジェクト解決を使用してテスト アプリケーションと対話します。動的オブジェクト解決により、オブジェクトを検索し識別する Silk Test Workbench クエリを使用した、テストの作成が可能になります。Silk Test Workbench は、ダイアログ ボックスやコントロールそれぞれに、固有に識別するためのロケーター名を割り当てます。Silk Test Workbench は、ロケーター名を使用してアプリケーション内のウィンドウ、ダイアログ ボックス、およびコントロールを検索します。同様に、Web ページにボタンなど多数のコントロールが含まれる場合があり、Silk Test Workbench はそれらにも固有のロケーター名を割り当てます。

Silk Test Workbench では、以下をテストできます。

- Silk Test とは異なるマシンで実行されるアプリケーション
- 2 台以上のマシンで同時に実行されるアプリケーション
- 複数の異なるアプリケーション (同時に)
- テスト アプリケーションのデータベース処理
- モバイル アプリケーション

Silk Test Workbench の UI

Silk Test Workbench のデスクトップは、テストにかかわるすべての操作の開始点になります。メニュー バーを除き、デスクトップに表示されるウィンドウやペインをはじめとする各種機能の表示と外観を制御することができます。

Silk Test Workbench UI の主要な構成要素を以下に示します。

メニュー Silk Test Workbench で利用できるすべてのメニューを含みます。利用可能なメニューとメニュー コマンドについての詳細は、「[メニュー](#)」を参照してください。

ツール ツール バーから、よく使用する操作にワンクリックでアクセスすることができます。ツール バーで利用可能なアクションについての詳細は、「[ツール](#)」を参照してください。

開始画面 **開始画面** は、Silk Test Workbench の機能を使い始める拠点であり、アプリケーションのテスト ソリューションをすばやく作成し始めることができます。詳細については、「[開始画面](#)」を参照してください。

アセット **アセットブラウザ** を使用してテスト資産を管理します。**アセットブラウザ** では、データベース内の各種類の資産を一元的に作成、管理、表示できます。詳細については、「[アセットブラウザ](#)」を参照してください。

ビジュアル **ビジュアル ナビゲータ** は、ビジュアル テストの要素をグラフィカルに表すものであり、ポイント アンド クリック インターフェイスを通して各要素を操作できます。**ビジュアル ナビゲータ** は、ビジュアル テストを作成、または開いたときに表示されます。詳細については、「[ビジュアル ナビゲータ](#)」を参照してください。

コード **コード ウィンドウ** では、スクリプトの記録、設計、および変更を行います。**コード ウィンドウ** は、VB .NET スクリプトを作成、または開いたときに表示されます。詳細については、「[コード ウィンドウ](#)」を参照してください。

Silk Test Workbench ソフトウェア コンポーネント

Silk Test Workbench は、別々のプロセスとして実行される 2 種類のソフトウェア コンポーネントから構成されます。

- *Silk Test Workbench* ホスト ソフトウェア は、スクリプトとビジュアル テストの開発、編集、コンパイル、実行、およびデバッグに使用するコンポーネントです。このプログラムを実行するマシンは、ホスト マシンと呼ばれます。
- *Silk Test Agent* は Silk Test Workbench のコンポーネントで、アプリケーションの GUI と対話します。Agent はテスト内のコマンドを GUI 固有のコマンドに翻訳し、テストしているアプリケーションを実行、監視します。Agent は、ホストが実行されているのと同じマシン上でローカルに実行できます。また、ネットワーク環境で任意の数のエージェントをリモート マシン上で実行することもできます。Agent を実行するマシンは、リモート マシンと呼ばれます。

自動テストについて

ソフトウェアのテストは、ソフトウェア開発プロセスに欠かせない要素です。ソフトウェアをリリースするには、その前に、ソフトウェアが期待どおりに動作することを、さまざまな種類のコンピュータやオペレーティング システムで確認する必要があります。その結果、ソフトウェアのテストはますます複雑化し、その機能をさまざまなコンピュータで繰り返しテストすることが必要になっています。しかし、ソフトウェアのテストが複雑になる一方で、テストに割り当てられる時間は、リリース スケジュールを守るように短縮されているのが実態です。テストを自動化すると、ソフトウェアにバグやパフォーマンス上の問題がないかどうかを効果的かつ効率的に調べることができます。そのため、自動テストツールは、高品質ソフトウェアを開発するうえで欠かせないツールとなっています。

アプリケーションを手動でテストするには、一連のテストケースを作成して、特定の操作（一連のキーを押す、メニュー オプションを選択する、マウスを使用するなど）を実行したときのテスト アプリケーションの必要な反応を指定します。手動によるテスト サイクルは、通常、以下のようなパターンに従います。

- テスト アプリケーションをテスト開始点まで実行する
- テスト アプリケーションに入力を提供する一連の操作を実行する
- アプリケーションと対話しながら結果を監視する
- 対話の結果と予想を比較する
- 予想と結果の違いを記録する
- 必要に応じてアプリケーションを修正する
- アプリケーションを再テストするか、次のテストに進む

Silk Test Workbench では、このテスト パターンの各ステップを模倣できます。つまり、キーボード入力を行い、マウスの移動やマウス ボタンのクリックを行い、コンピュータに表示される内容を予想と比較し、テストの結果を分析できるようファイルに保存することができます。テストをリンクしておく、1 つのテストが正しく終了したときに次のテストが開始されます。操作と検証が含まれるため、自動テストは手動によるテストより高速で、信頼性も高いものになります。

テスト戦略

ビジュアル テストやスクリプトなどの Silk Test Workbench 資産を作成してアプリケーションのテスト ソリューションを構築する前に、テスト戦略を立てることをお勧めします。

1 つのビジュアル テストまたはスクリプトに特定のテスト ソリューションのすべての部分を含める必要はありません。また、通常、そうすることは有益ではありません。

通常、最も効率のよいテスト方法は、モジュール式アプローチを採用したものです。アプリケーション テストを一連のトランザクション単位として考えます。

たとえば、オンライン発注システムのテストには、以下のようなトランザクション単位が含まれるかもしれません。

- オンライン システムへのログオン
- 顧客プロフィールの作成
- 注文の発注
- オンライン システムのログオフ

1つのテストでこれらの単位をすべて処理し、このテストを使用するシナリオが 10 通りある場合、それらのシナリオを処理するために 10 個の別々のテストを記録する必要があります。アプリケーションに何らかの変更があった場合、たとえば、ログオン ウィンドウにフィールドが 1 つ追加された場合、新しいフィールドへのデータ入力を処理するために 10 個の別々のテストに変更が必要になります。

これらのトランザクション単位をすべてテストするビジュアル テストまたはスクリプトを 1 つ作成して、それをシナリオごとに 10 個作成するよりも、これらのトランザクション単位を 1 つずつ処理する別個のテストをテスト「モジュール」として作成する方が有益です。トランザクション単位ごとに別々のテストを作成し、それをテスト シナリオごとに再利用すれば、ログオン トランザクション単位を処理するテストだけを変更すればよいことになります。

ドライバ スクリプトを実行すると、各テストが連続して呼び出されて実行されます。

モジュール式アプローチでは、在庫確認のテストを行う追加のテストを 1 つ作成し、それが適切な順序で呼び出されるようにドライバ スクリプトを更新できます。以下の例は、別々のスクリプトを呼び出してオンライン発注システムをテストするドライバ スクリプトを示しています。

```
Workbench.RunScript ("OnlineLogOn_Script")
Workbench.RunScript ("CreateProfile_Script")
Workbench.RunScript ("ValidateStock_Script")
Workbench.RunScript ("PlaceOrders_Script")
Workbench.RunScript ("OnlineLogOff_Script")
```

このようなモジュール式のテスト戦略を実施すると、テスト フローの変更も容易です。

たとえば、注文発注前の在庫確認など、オンライン発注システムのテストに追加のトランザクション単位が必要になった場合、10 個の異なるテスト シナリオがあれば、在庫確認を処理する 10 個のスクリプトを変更しなければなりません。

このようなモジュール式アプローチは、再利用性は向上しますが、毎回異なるデータを使用して一連のトランザクション単位を複数回繰り返すようなテスト戦略を要する、異なるテスト「シナリオ」に対応するには十分とは言えません。アクティブ データを使用すると、外部データ ソースを使用して、アプリケーション テストでのモジュール式テスト スクリプトの再利用を促進できます。

また、ソリューションをテストするためにドライバ スクリプトから関数を呼び出すこともできます。関数は、他のアプリケーション テストで再利用することができます。

別の場所にあるデータベースの利用について

分散環境で Silk Test Workbench を使用する場合、つまり、広域ネットワーク上で Silk Test Workbench クライアントから Silk Test Workbench の中央データベースにアクセスする場合、次の問題が発生する可能性があります。

- Silk Test Workbench クライアントとデータベース サーバー間での高レイテンシ。
- データベースへの多くのリクエストによるネットワーク トラフィックの増大。

このような分散環境では、Micro Focus では次のようなベスト プラクティスの適用をお勧めします。

- プロジェクト サイズを小さく保つ。これにより、**アセットブラウザ** の使用時などに、Silk Test Workbench クライアントとデータベース サーバー間でデータ転送量が増大するのを防ぎます。
- リモートでのテスト結果の表示を避ける。
- 可能な限り、中央データベースからプロジェクトをエクスポートしてオフラインで作業する。これを行うには、Silk Test Workbench ユーザーは、どの資産を読み取り専用として扱うか、どの資産を誰が更新するか、など調整する必要があります。さらに、中央データベースの資産をリモートのクライアントにインポートする必要があります。

- 仮想マシンやリモート デスクトップ接続を使用して、中央データベースにアクセスする。

Silk Test Workbench ドキュメント

Silk Test Workbench をマシンにインストールすると、(Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > ドキュメント**、または (Microsoft Windows 10) **スタート > Silk** をクリックしてドキュメントにアクセスできるようになります。

Silk Test Workbench ドキュメント セットには、以下のものが含まれています。

Silk Test インストールガイド	Silk Test Workbench のシステム要件とインストール手順について説明しています。さらに、基本的なセットアップと Silk Test Workbench データベースの作成、構成、および接続方法についても詳しく説明しています。Silk Test インストールガイドには、 オンライン ドキュメント からオンラインでアクセスすることもできます。
Silk Test リリースノート	新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『 リリース ノート 』を参照してください。
Silk Test Workbench ヘルプ	Silk Test Workbench 製品の使用方法に関する情報と Silk Test Workbench の言語リファレンスを提供します。
Silk Test チュートリアル	これらのチュートリアルには、特定のアプリケーション タイプを Silk Test でテストする方法や、Silk Test の特定の機能を使用する方法が記載されています。ローカルのインストール フォルダにあるチュートリアルのほか、最新のチュートリアルを オンライン ドキュメント からダウンロードすることもできます。

Silk Test Workbench の新機能

Silk Test Workbench では、以下の新しい機能がサポートされています。

HTML 形式の結果レポートにおけるイメージ検証

機能テストにおいて検証は重要な指標です。Silk Test では、アプリケーションの表示上の違いを比較して確認できます。アプリケーションの変更点やイメージ検証で失敗した理由を簡単に発見できるように、HTML 形式の結果レポートに、イメージの相違点を表示する機能が Silk Test 21.0 に追加されました。

ブラウザー サポートの自動更新

Chrome、Edge、Firefox、Opera などのブラウザーは、定期的かつ頻繁に更新されており、そのたびにすべてのテスト環境で Silk Test を手動で更新するのは非常に手間のかかる作業になっています。Silk Test 21.0 では、新しいバージョンのブラウザーをサポートするために必要なすべての WebDriver ファイルが自動的にダウンロードされるようになりました。隔離された環境で作業している場合は、内部のファイルストレージでダウンロードするファイルを管理することもできます。

スタンドアロン Recorder

Recorder や Locator Spy をコマンドラインから開始できるようになりました。これにより、Visual Studio Code などの IDE に記録機能をシームレスに統合できるようになりました。Recorder を使うと、Selenium ベースのスクリプト (Java、C#、VB) または Silk Test ベースのスクリプト (Java、C#) を生成できます。

AutoPass のサポート

Silk Test 21.0 では、Silk Meter ライセンス サーバーに加え、AutoPass ライセンス サーバーもサポートされるようになりました。

ユーザビリティの改善

Status ウィンドウの場所の設定

Silk Test Classic の Status ウィンドウの場所をカスタマイズできるようになりました。場所の設定は 5 つの解像度まで保存され、Silk Test Classic は適切なものを選択します。何も設定されていない場合は、デフォルトが使用されます。

VB.NET デバッグ時のグローバル オブジェクト

Silk Test では、.NET スクリプトのデバッグ時にグローバル オブジェクト (モジュール レベルで宣言されたオブジェクト) が表示されるようになりました。

使用技術の更新

Silk Test は、Microsoft Edge (Chromium ベース) 81 から 92 をサポートするようになりました。

Silk Test は、Java 14、15 をサポートするようになりました。

Silk Test は、SAPGUI クライアント 7.60 をサポートするようになりました。

Silk Test は、iOS 14 と Android 11 をサポートするようになりました。

プロジェクトの管理

プロジェクトは資産のコレクションであり、各プロジェクトではアクセス権限を各ユーザーに割り当てます。Silk Test Workbench 環境の管理者が行うタスクの 1 つに、プロジェクト リストの管理があります。資産を論理的に関連付けたプロジェクトとしてまとめて保管します。たとえば、特定のアプリケーションまたはビルドに関連付けられた資産をすべて一緒に、1 つのプロジェクトに保管できます。

管理者は、プロジェクトを作成して、Silk Test Workbench データベースに追加できます。作成した新しいプロジェクトのアクセス権は、各ユーザーに設定された [デフォルトの権限] が割り当てられます。資産を別々のプロジェクトに保管すると、セキュリティのレベルを変えることができます。たとえば、プロジェクト 1 はフル アクセス権、プロジェクト 2 は読み取り専用、プロジェクト 3 はアクセス不可、のように設定できます。

現在のユーザーには、アクセス権を割り当てられているプロジェクトの資産だけが表示されます。資産は、**アセット ブラウザ** を使用して表示および管理します。




注: Silk Test Workbench が提供する機能を最適に使用するには、同じテストで複数のアプリケーションをテストする場合を除き、テストするアプリケーションごとに個別のプロジェクトを作成します。

Common プロジェクト

デフォルトのプロジェクトは Common プロジェクトです。これは、複数のプロジェクトで使用される資産の中央リポジトリとして機能します。Common プロジェクトの資産は、他のすべてのプロジェクトで使用できます。Common プロジェクトはデフォルトでアクティブ プロジェクトになっています。

アクティブプロジェクト

新しく作成される資産は、アクティブプロジェクトとして設定されているプロジェクトに追加されます。アクティブに追加された資産は、そのプロジェクト内と、アクティブプロジェクトを参照しているすべてのプロジェクト内でのみ使用可能になります。

 **注:** プロジェクトの資産は、同じプロジェクトまたは参照プロジェクトの他の資産にのみアクセスできます。

プロジェクトの表示

アセットブラウザ では、**表示** ダイアログ ボックスで選択したプロジェクトとアクティブプロジェクトに属するテスト資産がすべて表示されます。アクセスできるプロジェクトのテスト資産のみを表示できません。

デフォルトでは、新しい資産は常にアクティブプロジェクトに追加され、そのプロジェクトでだけ使用できます。

1. **表示** > **アセットブラウザ** を選択するか、**開始画面** の **入門ガイド** > **タスク** に移動して、**アセットブラウザ** をクリックします。**アセットブラウザ** が開きます。
2. アクティブプロジェクトを変更するには、**アセットブラウザ** ツールバーの **表示** をクリックします。**プロジェクトの資産の表示** ダイアログ ボックスが開き、現在のユーザーがアクセスできるプロジェクトのリストが表示されます。
3. 表示する資産を持つプロジェクトのチェック ボックスをオンにし、**OK** をクリックします。**アセットブラウザ** には、選択したすべてのプロジェクトに含まれているテスト資産が表示されます。アクティブプロジェクトの資産は常に表示されます。

ショートカット キー

標準の Windows メニュー コマンド以外に、多くのコマンドにキーの組み合わせが関連付けられており、キー入力によってコマンドを実行できます。通常、Ctrl または Alt キーを押しながら、該当するコマンドメニューの下線付きの文字を押します。

コマンドのメニュー ショートカット キー

ショートカット キー	実行される操作
Alt+F9	選択したオブジェクトを使用して、記録中にビジュアル テストまたはスクリプトに検証を挿入します。
Alt+F10	記録を開始または停止します。
Alt+Enter	ビジュアル テストまたはスクリプトから 要約情報 ダイアログ ボックスを表示します。
Ctrl+Alt	ビジュアル テスト (記録または再生中) またはスクリプト (記録中) に検証を挿入します。
Ctrl+F8	現在選択されているテスト ステップまでビジュアル テストを再生するか、または現在選択されているステートメントまでスクリプトを再生します。
Ctrl+F9	次のステートメントまたはポイントまで再生を実行します。デバッグ モードの場合に有効になります。
Ctrl+K	スクリプトで現在選択されている行 (複数) をコメントアウトして無効化します。コードブロック全体をコメントアウトするために、このショートカットを使用できます。また、 Ctrl+F11 を使用することもできます。
Ctrl+N	新規作成 ウィンドウを表示し、任意の種類 of 資産を作成することができます。

ショートカット キー	実行される操作
Ctrl+O	アセットブラウザ を開き、資産を開くことができます。前回作業した資産の種類が強調表示されます。
Ctrl+P	[印刷] ダイアログ ボックスを表示して、選択した資産を印刷できます。
Ctrl+S	<ul style="list-style-type: none"> 資産の現在の状態を資産の新しいバージョンとして保存し、増分の数値識別子をそのバージョンに割り当てます。ショートカット キーの組み合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。 資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
Ctrl+Y	直前の 元に戻す 操作をやり直します。資産の比較、またはマージ時には利用できません。
Ctrl+Z	直前の編集を元に戻します。資産の比較、またはマージ時には利用できません。
Ctrl+Alt+A	アセットブラウザ を開きます。
Ctrl+Alt+I	選択したオブジェクトのロケーターを記録できる オブジェクトの識別 ダイアログ ボックスを開きます。
Ctrl+Alt+S	開始画面 を表示します。
Ctrl+Shift+F2	前回の位置を表示します。
Ctrl+Shift+F8	別のビジュアル テストから再生しているビジュアル テスト内の残りのステップをすべて実行したあと、元のビジュアル テスト内の次のステップで再生が一時停止します。スクリプトの場合は、プロシージャ内の残りすべてのコードを 1 つのステートメントのように実行し、そのプロシージャを最初に呼び出したプロシージャ内の次のステートメントに移ります。
Ctrl+Shift+F9	すべての開いているビジュアル テストと VB .NET スクリプトのすべてのブレークポイントをクリアします。
Ctrl+Shift+K	スクリプトで現在選択されている行 (複数) のコメントを解除して有効化します。
Ctrl+Shift+L	<ul style="list-style-type: none"> 開いているすべての資産の現在の状態を資産の新しいバージョンとして保存し、各資産のバージョン識別子をインクリメントします。ショートカット キーの組み合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。 開いているすべての資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
Ctrl+Tab	ユーザー インターフェイスで開いているウィンドウを切り替えます。
Ctrl+Space	単語を補完します。
Del	選択した資産を削除します。
F1	特定の項目のヘルプを表示します。
F2	プロパティ ペインで選択したパラメータの種類に応じて、 スクリプト入力パラメータの編集 または スクリプト出力パラメータの編集 ダイアログ ボックスを表示します。
F4	プロパティを表示します。

ショートカット キー	実行される操作
F5	アクティブなビジュアル テストまたはスクリプトを再生します。
F7	スクリプトの構文がコンパイルのすべての条件を満たしているかをチェックし、コードをコンパイルし、エラーが発生した場合は表示します。
F8	ビジュアル ステップの場合、テスト ステップを一度に 1 つずつ再生します。スクリプトの場合、ステートメントを一度に 1 つずつ再生します。
F9	選択したステップまたはコード行にブレークポイントを設定するか、または選択したステップまたはコード行に設定されているブレークポイントをクリアします。
Shift+F2	定義を表示します。
Shift+F8	埋め込まれているビジュアル テストまたはスクリプト全体を再生し、元のビジュアル テストまたはスクリプト内の次のステップで停止してデバッグ モードになります。VB .NET スクリプトで押下された場合は、関数呼び出しのステップ オーバーも行います。

チュートリアル

Silk Test Workbench チュートリアルでは、Silk Test Workbench を使用してアプリケーションをテストする基本的な方法をご自身のペースで学習することができます。また、チュートリアルでは、テスト アプリケーションのさまざまな条件を処理するパワフルで柔軟なテストを作成したり、再生中に発生した詳細を正確にレポートしたりするための高度な機能もいくつか紹介しています。

チュートリアルではハンズオン デモが提供されており、Silk Test Workbench と直接対話しながら現実の世界のシナリオを使ってサンプル アプリケーションをテストします。



注: Silk Test Workbench チュートリアルで使用されているサンプル アプリケーションは、Internet Explorer に合わせて設計され、最適化されています。チュートリアルのレッスンと同じユーザー体験を得るには、Internet Explorer 以外のサポート対象ブラウザでチュートリアル サンプル アプリケーションを実行しないでください。

Silk Test Workbench ビジュアル テスト チュートリアル

Silk Test Workbench ビジュアル テスト チュートリアルによろこそ。ご自身のペースで学習できるこのチュートリアルでは、Silk Test Workbench のビジュアルでストーリーボードベースのインターフェイスを使用して、パワフルで柔軟性の高い機能テストを作成する方法について説明します。このチュートリアルでは、ビジュアル テストの作成、ビジュアル テストの再生、および再生結果の分析に必要な基本ステップについて学習します。また、多数の機能を使用して、記録したビジュアル テストをすばやく更新および拡張する方法についても学習します。

このチュートリアルでは、Silk Test Workbench を使用して繰り返し可能なテストを作成する練習をするための Silk Test サンプル Web アプリケーション <http://demo.borland.com/InsuranceWebExtJS/> を使用します。

各レッスンは前のレッスンの出力に基づいて進められるため、チュートリアルのレッスンは順番に完了してください。

GUI の概要

このセクションでは、メイン画面、開始画面、およびビジュアル ナビゲータを含めた GUI について説明します。このセクションはオプションです。すでに開発環境の基本要素をご存知の場合は、次のレッスンに進んでください。

Silk Test Workbench のメイン画面

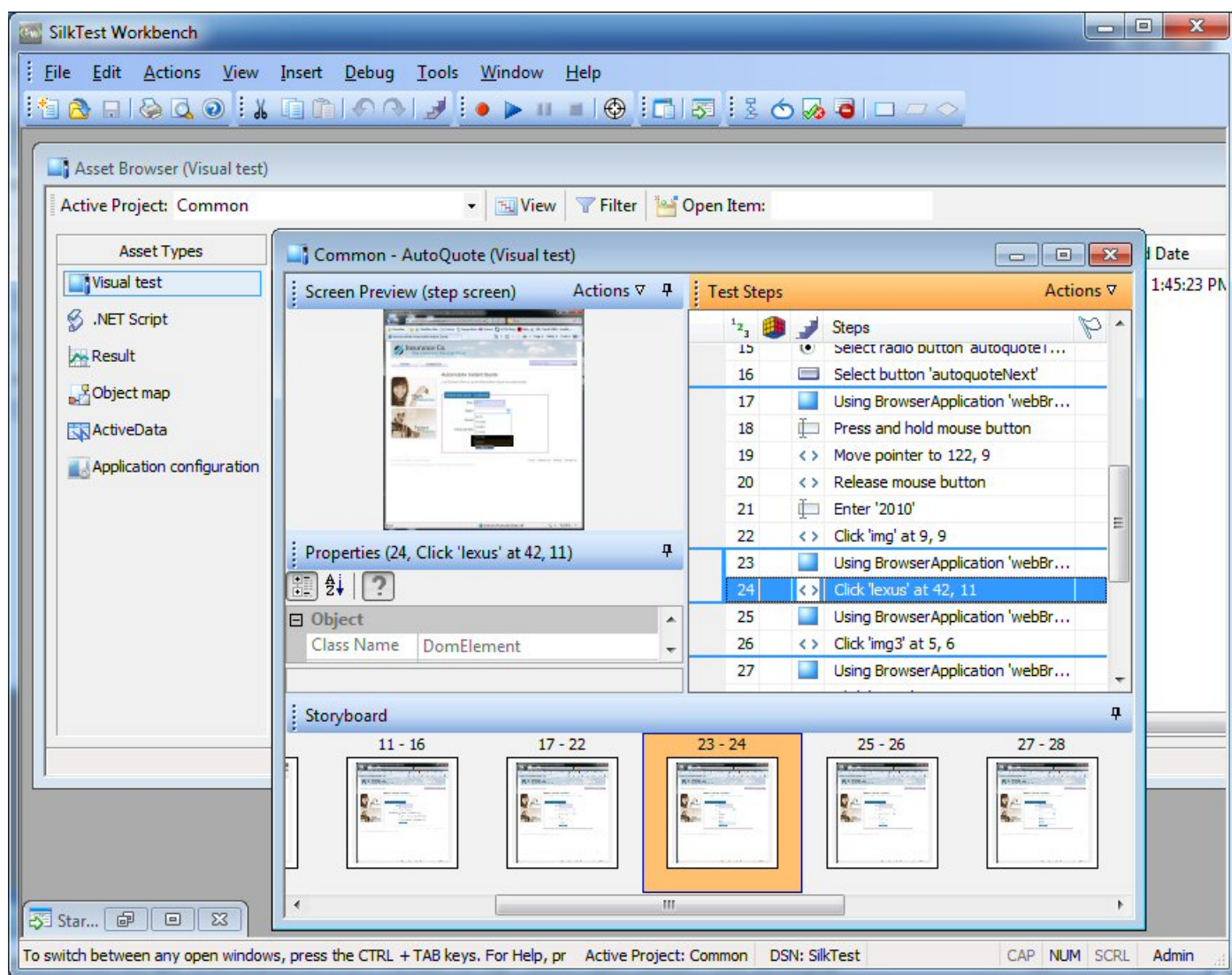
Silk Test Workbench のメイン画面は、テストにかかわるすべての操作の開始点になります。メニューバーを除き、デスクトップに表示される項目はすべてユーザーが制御できます。

開始画面、ビジュアルナビゲータ、アセットブラウザなど、Silk Test Workbench のその他すべてのウィンドウは、Silk Test Workbench のメイン画面の子ウィンドウとして表示されます。

Silk Test Workbench のメイン画面には以下の要素が含まれます。

- メニューバー
- ツールバー
- メインのテスト資産表示領域
- ステータスバー

以下の図は、Silk Test Workbench のメイン画面のこれらの要素を示しています。メインのテスト資産表示領域では、ビジュアルナビゲータにビジュアルテストが表示され、背景にアセットブラウザが表示され、開始画面が最小化されて表示されています。



開始画面

開始画面 は、Silk Test Workbench の機能を使い始める拠点であり、アプリケーションのテストソリューションをすばやく作成し始めることができます。

アプリケーションのテストをすばやく、最低限の手順で開始できるようにするため、**開始画面**には次の領域が含まれています。

領域 **説明**

開始 ビジュアルテスト、.NET スクリプト、キーワード駆動テストをワンクリックで記録したり、開いたりできます。

資産 最近使ったビジュアルテスト、.NET スクリプト、キーワード駆動テストをワンクリックで開くことができます。最近使った資産のリストは、リアルタイムで更新されるので、変更や更新により、現在の資産がリストの先頭に移動します。資産の右側にあるアイコンをクリックすることによって、資産を再生したり、資産の最後の実行結果を表示したりすることができます。



注: 開始画面 設定 (ツール > オプション > 開始画面) では、最近使った資産のリストに表示する項目の数を制御できます。

新機能 Silk Test Workbench の現在のバージョンに含まれている新しい機能や機能強化を一覧します。Silk Test Workbench の更新が利用可能な場合も、ここに表示されます。



注: Silk Test Workbench の起動時に開始画面を表示しないように選択している場合、ヘルプ > 製品更新の確認 をクリックして利用可能な更新を確認できます。

入門ガイド 生産性向上のため、広く使われるタスクや有用なヘルプ トピックへの直接リンクを提供します。

リソース サポート技術情報や Silk Test コミュニティなどの Silk Test Workbench に関連した情報へのリンクを提供します。

お知らせとコミュニティ Silk Test コミュニティの最新のフォーラム記事とブログ投稿へのリンクを提供します。

フラグ ビジュアルテストのテスト ステップ、およびフラグが付けられてフォローアップ対象として割り当てられた結果が表示されます。**フラグ** ペインを使用すると、テスト スイートやテスト プロジェクトについての情報を交換でき、データベースのテスト プロジェクトのほかのテスト担当者やユーザーと共同作業が行えます。

デフォルトでは、フラグは作成日および表示される資産で分類されます。フラグの情報をすばやく表示するには、マウスのポインタをフラグの上に移動します。フラグの説明、更新データ、プロジェクト、フラグが割り当てられているユーザーの ID を示すツール ヒントが表示されます。**開始画面のフラグ オプション (ツール > オプション > 開始画面 > フラグ)** を変更して、フラグの表示方法を変更できます。

フラグに関連付けられているビジュアル テストまたは結果を開くには、**フラグ** ペインでフラグをダブルクリックします。



注: フラグ ペインは、少なくとも 1 つのフラグがログインしているユーザーに割り当てられているか、少なくとも 1 つのフラグがフィルタで表示されるように設定されている場合にのみ、**開始画面** に表示されます。

開始画面 の 右上隅にあるウィンドウの **閉じる** をクリックすると、**開始画面** を閉じることができます。



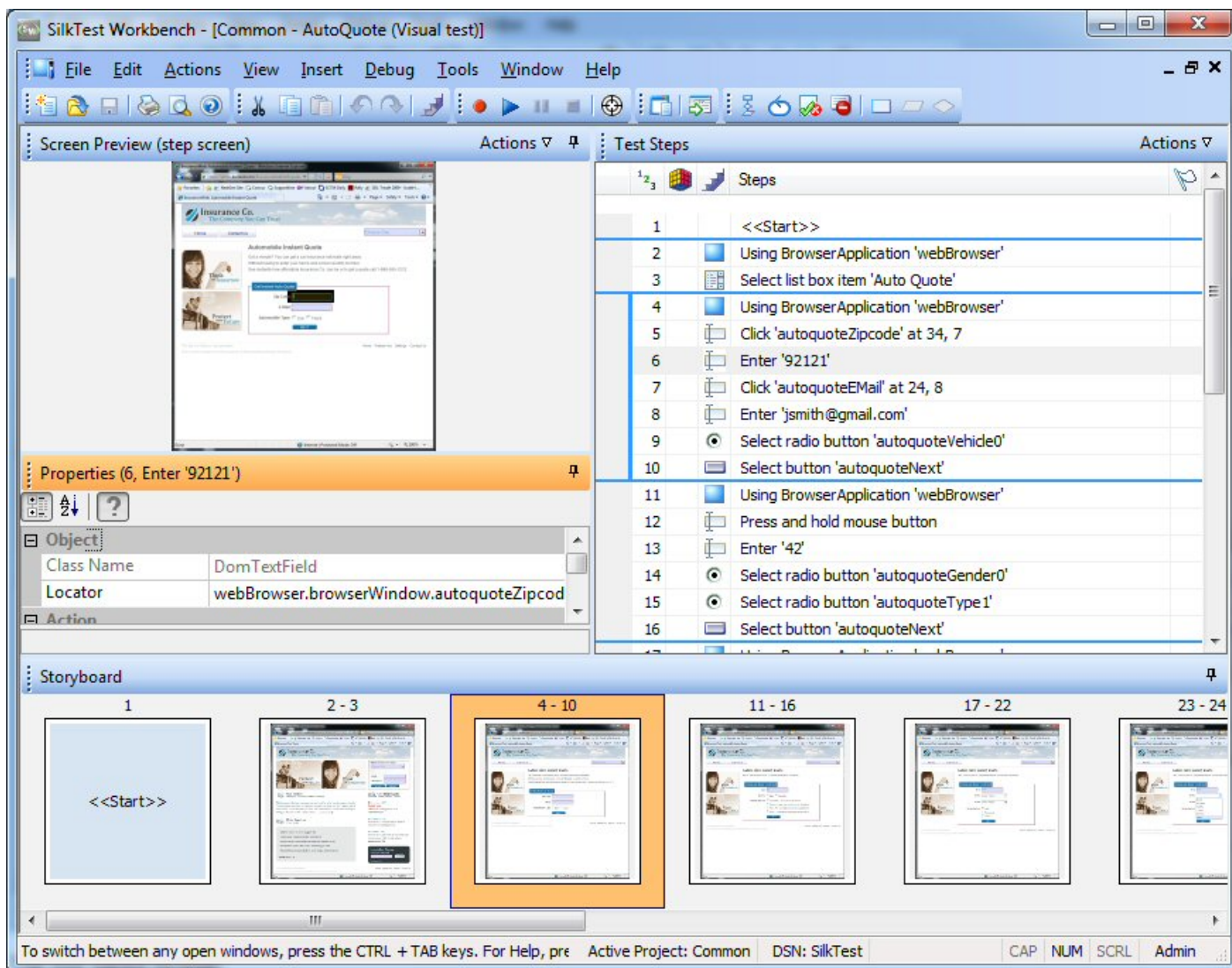
ヒント: ツールバーの **開始画面** をクリックすることで、メイン画面から **開始画面** にいつでもアクセスできます。また、いつでも Ctrl+Alt+S を押すか、または **表示 > 開始画面** を選択して、開始画面を表示できます。



注: ユーザー インターフェイスは、Windows 標準の大きいフォントや小さいフォントを使用します。カスタムのフォント サイズを使用すると、ユーザー インターフェイスのテキスト表示が不正確になる可能性があります。

ビジュアルナビゲータ

ビジュアルナビゲータは、ビジュアルテストの要素をグラフィカルに表すものであり、ポイントアンドクリックインターフェイスを通して各要素を操作できます。**ビジュアルナビゲータ**では、ビジュアルテストに関する情報が4つのペインに表示されます。4つのペイン全体でビジュアルテストの各ステップを総合的に把握できます。



4つのペインは以下のとおりです。

テストステップ ビジュアルテストの各ステップが技術的ではない明確な言語で示されます。

画面プレビュー ビジュアルテストの再生中に実行されるステップに応じてテスト対象アプリケーションのスナップショットが表示されます。

プロパティ ビジュアルテストのステップのプロパティが表示されます。

ストーリーボード サムネイルイメージを使用してビジュアルテストのフローが表示されます。サムネイルは、ビジュアルテストのステップの論理グループを表します。

 **注:**

Silk Test Workbench では、スナップショットは以下のような状況のときに取得されます。

- 記録中のすべての自動テスト ステップの前。



注: SAP アプリケーションの場合、スナップショットはすべての自動テスト ステップの前ではなく、画面が変わったときに発生します。

- ビジュアル テストで Using ステップを実行した結果のスナップショット。
- 再生エラーが発生した場合。

画面プレビュー、**ストーリーボード**、および**プロパティ** ペインは **テスト ステップ** ペインと同期され、**テスト ステップ** ペインで選択したステップに関する情報が表示されます。このように、**テスト ステップ** ペインでステップを選択し、他のペインでそのステップに関する情報を確認することで、ステップのすべての側面を簡単に把握できます。

ビジュアル ナビゲータ では、ビジュアル テストを表示するだけでなく、**画面プレビュー** ペインと**プロパティ** ペインを使用して既存のビジュアル テストを拡張または更新することもできます。たとえば、ビジュアル テストを記録したあと、**プロパティ** ペインで、記録したプロパティのリテラル値を変数に置き換えることができます。また、テスト対象アプリケーションに変更が発生したときにビジュアル テストをすばやく更新する場合は、**画面プレビュー** の **画面の更新** 機能を使用して、以前にキャプチャした画面を更新できます。

ビジュアル ナビゲータ では、ビジュアル テストに使用するのと同じペインを使用して、ビジュアル テストの再生結果も表示します。結果の場合、これらのペインには機能が付加され、**結果** ウィンドウ内に表示されます。ウィンドウにはツールバー オプションと複数のタブが表示されます。各タブには結果のさまざまな内容が表示されます。結果に固有の追加機能として、たとえば、**テスト ステップ** ペインに各ステップのステータス（合格または失敗）を表示する機能があります。さらに、**画面プレビュー** では、記録中にキャプチャされた画面と再生中にキャプチャされた画面の違いを比較し、テスト アプリケーションにアクセスすることなく、既存のビジュアル テストを更新できます。

ビジュアル テストの記録開始画面

サンプル Web アプリケーションで保険の見積もりを作成するための操作を行うと、Silk Test Workbench はそれらの操作をステップとして記録します。ステップはビジュアル テストの基礎となります。テストに必要な操作の記録を完了すると、ビジュアル ナビゲータに記録されたテストが表示されます。ステップは、ビジュアル ナビゲータの [テスト ステップ] ペインに表示されます。



注: このチュートリアルで使用されているサンプル アプリケーションは、Internet Explorer に合わせて設計され、最適化されています。チュートリアルのレッスンと同じユーザー体験を得るには、Micro Focus は Internet Explorer 以外のサポート対象ブラウザでチュートリアル サンプル アプリケーションを実行することをお勧めしません。



注: Web アプリケーションを記録または再生する前に、システムにインストールされているすべてのブラウザアドオンを無効にします。Internet Explorer でアドオンを無効にするには、**ツール > インターネット オプション** をクリックし、**プログラム** タブをクリックし、**アドオンの管理** をクリックし、アドオンを選択してから **無効にする** をクリックします。

サンプル Web アプリケーションの起動

このチュートリアルでは、Silk Test サンプル Web アプリケーションを使用します。この Web アプリケーションは、デモ用に提供されています。

Internet Explorer で、Silk Test サンプル Web アプリケーションを使用します。チュートリアルのレッスンと同じユーザー体験を得るには、Internet Explorer 以外のサポート対象ブラウザでサンプル Web アプリケーションを実行しないでください。

1. テストの速度と信頼性を高めるために DOM 関数を記録するには、以下のステップを実行します。
 - a) **ツール > オプション** をクリックします。
 - b) **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。

- c) **xBrowser** をクリックします。
- d) **ネイティブなユーザー入力を記録する** リスト ボックスから、**いいえ** を選択します。
- e) **OK** をクリックします。



注: 通常、Web アプリケーションをテストする場合、DOM 関数ではなく、ユーザーの入力そのものを使用します。Flash および Java アプレットなどのプラグインや、AJAX を使用するアプリケーションは、ネイティブなユーザー入力ではサポートされますが、高レベルの API 記録ではサポートされません。

2. Web アプリケーションの記録または再生を行う前に、ブラウザのアドオンをすべて無効にする必要があります。ブラウザのすべてのアドオンを確実に無効にするには、以下のステップを実行します。
 - a) Internet Explorer で**ツール > インターネット オプション** を選択します。**インターネット オプション** ダイアログ ボックスが開きます。
 - b) **プログラム** タブをクリックし、**アドオンの管理** をクリックします。**アドオンの管理** ダイアログ ボックスが開きます。
 - c) アドオンのリストの**状態** 列で、各アドオンの状態が**無効** になっていることを確認します。**状態** 列に**有効** と表示されている場合は、アドオンを選択してから**無効にする** をクリックします。
 - d) **閉じる** をクリックし、**OK** をクリックします。
3. サンプル アプリケーションにリモートでアクセスするには、<http://demo.borland.com/InsuranceWebExtJS> をクリックします。サンプル アプリケーションの Web ページが開きます。

サンプル Web アプリケーションに対するビジュアル テストの記録

記録中は、記録を停止するまで、テスト アプリケーションでのすべての操作（Silk Test Workbench の操作は除く）が Silk Test Workbench によって記録されます。記録が完了したら、生成したビジュアル テストを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. このチュートリアルの後半で記録したコントロールを編集できるようにするために、ビジュアル テストを記録する前にブラウザ コントロールのキャプチャを有効化します。
 - a) Silk Test Workbench メニューで、**ツール > オプション** を選択します。**オプション** ダイアログ ボックスが開きます。
 - b) ツリーから**記録 > 出力 > ビジュアル テスト** を選択します。
 - c) **ブラウザ コントロールのキャプチャ** の値を**はい** に設定します。



注: Web アプリケーションの記録中にコントロールのキャプチャを有効化すると、ビジュアル テストのパフォーマンスが低下する恐れがあります。Micro Focus では、このチュートリアルで述べるタスクを完了したら、コントロールのキャプチャを無効化することをお勧めします。

2. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
3. 資産の種類のリストから**ビジュアル テスト** を選択します。
4. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
5. **OK** をクリックして、ビジュアル テストを資産として保存し、記録を開始します。**アプリケーションの選択** ダイアログ ボックスが開きます。
6. **Web** タブを選択します。
7. リストから**Internet Explorer** を選択します。
8. **移動する URL の入力** テキスト ボックスに、demo.borland.com/InsuranceWebExtJS/ を入力します。
9. **OK** をクリックします。Silk Test Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。
10. Insurance Company Web サイトでは、次のステップのいずれかを行います：
 - a) **Select a Service or login** リスト ボックスから**Auto Quote** を選択します。**Automobile Instant Quote** ページが開きます。
 - b) 郵便番号と電子メール アドレスを適切なテキスト ボックスに入力し、自動車タイプをクリックして、**Next** をクリックします。
たとえば、郵便番号に 92121、電子メール アドレスに jsmith@gmail.com をそれぞれ入力し、自動車タイプとして Car を指定します。

- c) 年齢を指定し、性別と運転履歴タイプをクリックして、**Next** をクリックします。
たとえば、年齢に 42 を入力し、性別と運転履歴タイプに Male および Good をそれぞれ指定します。
- d) 製造年、車種、モデルを指定し、財務情報タイプをクリックして、**Next** をクリックします。
たとえば、製造年に 2010 と入力し、車種とモデルに Lexus および RX400 をそれぞれ指定し、財務情報タイプとして Lease を指定します。
指定した情報の概要が現れます。
- e) **Purchase** をクリックします。
Purchase A Quote ページが開きます。
- f) ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。

11 Alt+F10 を押すか、**記録中** ウィンドウで **停止** をクリックするか、Silk Test Workbench のタスクバーアイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェックボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、ビジュアル テストは **テストステップ** ペインに表示されます。

ビジュアル テストの保存と名前の指定

テストを記録すると、**記録完了** ダイアログ ボックスが開き、以下の操作を実行できます。

- 記録したビジュアル テストを再生する
- 記録した操作をビジュアル ナビゲータで確認する
- ビジュアル テストを保存し、記録したテストをビジュアル ナビゲータで確認する

ビジュアル テストをはじめて記録したばかりなので、テストを再生したり、記録した操作を確認したりする前にテストを保存し、名前を付けます。

1. **記録完了** ダイアログ ボックスで、**保存** をクリックします。



ヒント: 名前を指定せずに資産を作成すると、Silk Test Workbench により「無題_」に連番の付いた一時的な名前が割り当てられます。

テストにまだ名前を付けていないため、**名前を付けて保存** ダイアログ ボックスが開きます。

2. **名前** テキスト ボックスで、名前を AutoQuote に変更します。

3. **説明** テキスト ボックスに Visual test tutorial と入力します。

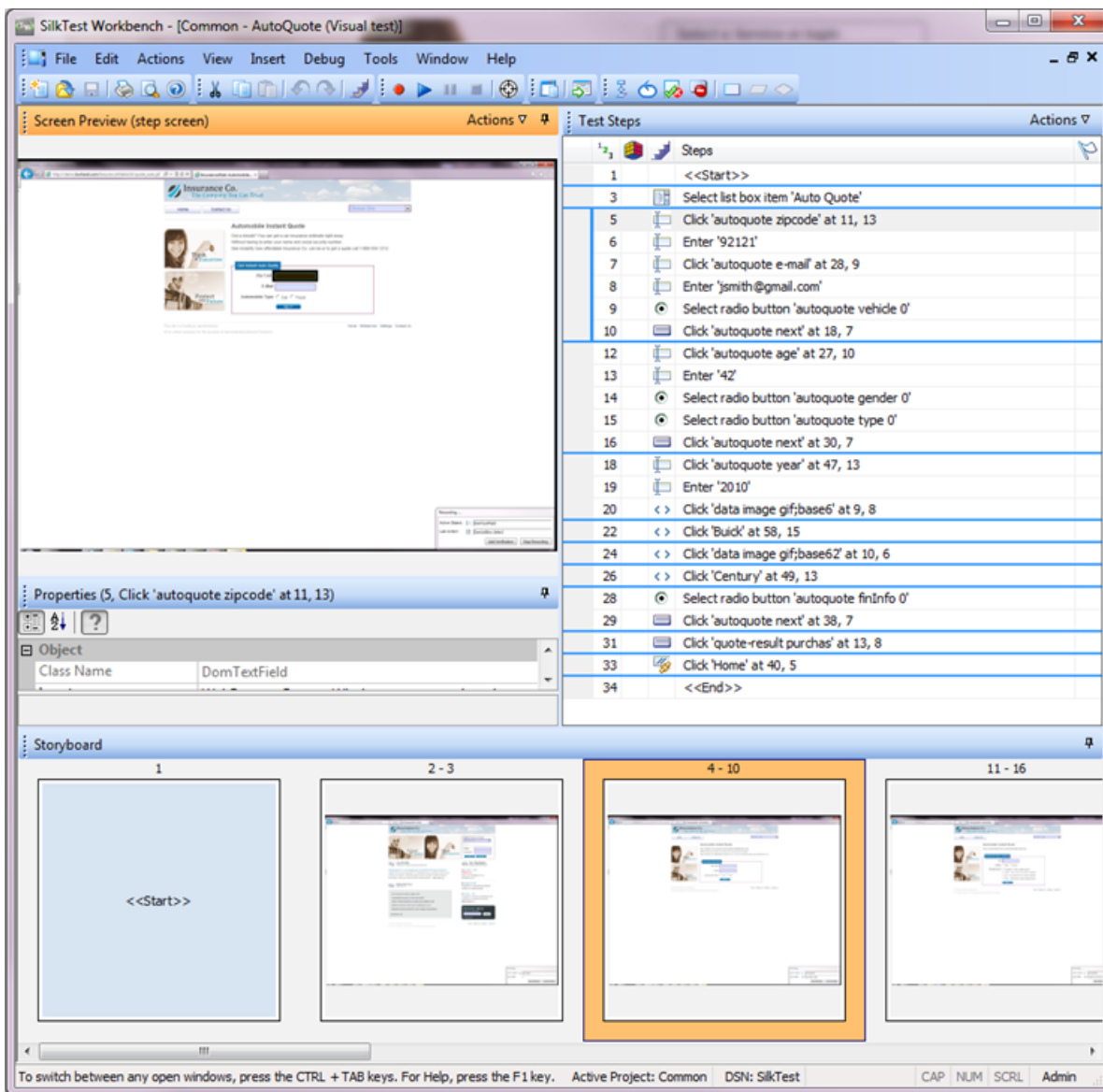
4. **OK** をクリックします。ビジュアル ナビゲータにビジュアル テストが表示されます。

記録されたテスト ステップの確認





ビジュアル テストを記録し、保存すると、ビジュアル テストがビジュアル ナビゲータに表示されます。ビジュアル ナビゲータには 4 つのペインがあり、ビジュアル テストの包括的なビューが提供されます。


[テスト ステップ] ペインのステップは、Web アプリケーションで見積もりを完了するときにアクセスした画面と実行した操作を表します。


記録されたステップは、以下の図に示すステップと同じようになるはずですが。



[テスト ステップ] ペインには以下の列が表示されます。

列	説明
 (番号) *	ステップが記録され、再生される順番を表します。デフォルトでは、ステップはこの順番で表示されます。 ステップにデバッグのためのブレークポイントが追加されている場合は、この列の隣の列にブレークポイントが表示されます。 デフォルトのビュー（ステップと画面）から画面のみのビューまたはステップのみのビューにビューを変更した場合、番号付けの方法では、記録の順序で間隔が示されるようにステップまたは画面は順番にスキップされます。
 (ロジック) *	ステップにロジックが含まれている場合、ステップのロジックのタイプを表すアイコンを表示します。
 (ステップ タイプ) *	ステップで実行される操作のタイプを表すアイコンを表示します。
ステップ	ステップで実行される操作について説明します。
 (フラグ) *	割り当てられたフラグ のアイコンを表示します。

列	説明
 (ステップの説明) *	<p>割り当てられたフラグ のアイコンは、関連付けられたステップが割り当て先のユーザーの テストステップ ペインおよびオプションで 開始画面 ペインに表示されるように、フラグが付けられていることを示します。さらに、ポインタをアイコンの上に移動すると、フラグの説明、更新日、および割り当て日を示すツールヒントを表示できます。</p> <p>ユーザー定義のステップの説明を表示します。この列はデフォルトのビューには表示されません。テストステップ ペインのタイトルバーから 操作 をクリックし、次に 表示 > ステップの説明 をクリックすると表示できます。</p> <p>ステップの説明を作成するには、ステップを選択します。プロパティ ペインで、ステップの説明 プロパティを更新します。ステップの説明 アイコンがこの列に表示され、ステップに説明があることが示されます。ポインタをアイコンの上に移動して説明を含むツールヒントを表示するか、ステップを選択してプロパティ ペインの説明を読みます。</p>

 **ヒント:** ビジュアルナビゲータの他のペインは、**テストステップ** ペインと同期しています。上の図では、記録したステップのうち、リストボックスから **Auto Quote** を選択するステップが **テストステップ** ペインで選択されています。結果は以下のようになります。

- **画面プレビュー** には、**Auto Quote** を選択する前のアプリケーションの状態が示されます。
- **プロパティ** ペインには、選択されたステップのプロパティが示されます。
- ストーリーボードでは、**Auto Quote** リスト項目の選択に関連するステップのグループを表すサムネイルが強調表示されます。

ビジュアルテストのステップをスクロールし、さまざまなステップを選択すると、他のペインに更新した情報が表示されます。

記録されたビジュアルテストの再生

ビジュアルテストを記録し、保存したら、再生して、ビジュアルテストが適切に動作することを確認できます。

1. 次のいずれか 1 つのステップを行います：

- **操作 > 再生** を選択します。
- ツールバーで **再生** をクリックします。

再生 ダイアログボックスが開きます。このダイアログボックスで、結果をどのように保存するかを指定できます。

2. **結果の説明** テキストボックスに Initial test results for the recorded test と入力します。

3. **OK** をクリックします。

4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログボックスが開きます。ブラウザーを選択して、**実行** をクリックします。

各結果は、一意のテスト実行番号によって識別されます。

Silk Test Workbench が最小化され、ビジュアルテストが再生されます。再生中、サンプルアプリケーションアプリケーションに対して、ビジュアルテストの記録中に行った操作が画面上で再生されます。再生が正常に完了すると、**再生完了** ダイアログボックスが開きます。

結果の分析：概要

テストを再生すると、Silk Test Workbench によってテスト結果が生成されます。テスト結果にはビジュアルテストの再生に関する情報が含まれています。ビジュアルテストの名前、実行番号、各ステップが実行された日付と時刻、各ステップの合格/失敗のステータスなどの重要な情報が含まれます。

[結果] ウィンドウのタブの使用

テスト結果情報をすばやく参照するために、**結果** ウィンドウには、結果情報を種類別に分けて表示するフィルタの役目を果たす 5 つのタブがあります。

1. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。

結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。**要約** タブには、再生が成功したかどうか、最新の実行番号、合格または失敗となったチェックまたは検証の数、テストの開始時刻と終了時刻、テストの実行結果に関するその他の基本情報などを含むテスト実行の概要が示されます。

2. **詳細** タブをクリックします。

詳細 タブでは、ビジュアル ナビゲータの以下の 4 つのペインを使用して各ステップの結果が示されます。

テスト ステップ ビジュアル テストの各ステップの再生結果に関する情報がリストされます。

画面プレビュー 再生中にキャプチャされた Web アプリケーションの画面が表示されます。

プロパティ ステップのプロパティが表示されます。

ストーリーボード 結果のアウトラインと概要がグラフィカルに示されます。



注: 合格、失敗、フラグ タブでもビジュアル ナビゲータを使用して結果情報が示されます。違いは、**詳細** タブではすべてのステップが示されるのに対し、これらのタブでは特定の種類のステップが示されるという点だけです。

3. 任意のステップを選択します。

ステップを選択すると、Silk Test Workbench により他のペインの情報がそのステップの情報に更新されます。**画面プレビュー** では、再生中にキャプチャされた画面が、ビジュアル テストを最初に記録したときにキャプチャされた画面と比較されます。**プロパティ** ペインには、選択したステップのプロパティが示されます。**ストーリーボード** では、ステップが属するグループが強調表示されます。



ヒント: ステップの名前全体を表示するには、**ステップ** 列を広げるか、ステップの上にマウス ポインタを置いて、名前全体を示すツール ヒントを表示する必要がある場合があります。

[結果] ウィンドウのツールバーの使用

結果 ウィンドウのツールバーには、結果の内容の表示と種類をカスタマイズするためのいくつかのオプションがあります。

1. 結果のツールバーの **詳細表示** をクリックします。Silk Test Workbench により、**テスト ステップ** ペインに追加の列が表示され、**プロパティ** ペインに追加のプロパティが表示されます。
2. **テスト ステップ** ペインで、右にスクロールして追加の列を表示します。
これらの列には、ステップの実行に要したミリ秒単位の時間、テストを実行した人のユーザー名など、各ステップに関する特定の情報が示されます。
3. **プロパティ** ペインの [結果] プロパティ カテゴリには、**テスト ステップ** ペインの各列に対応するプロパティが示されます。下にスクロールして、リスト全体を表示します。
4. **表示** ドロップダウン矢印をクリックし、**ステップのみ** を選択します。Silk Test Workbench によりすべての画面が除外されるので、ステップだけをすばやく参照できます。
5. **表示** ドロップダウン矢印をクリックし、**ステップと画面** を選択します。Silk Test Workbench により結果の画面とステップがすべて表示されます。

結果 ウィンドウのツールバーには、この他にも結果の内容の表示と種類をカスタマイズするためのオプションがあります。

[プロパティ] ペインの使用

プロパティ ペインには、ステップの基本特性を示すプロパティが表示されます。ステップの名前、実行ステータス、再生の詳細などの情報が示されます。

1. ステップ「'42' を入力します」をクリックします。

このステップは、**Age** テキスト ボックスに値「42」を入力する操作を実行します。

プロパティ ペインが更新され、選択したステップのプロパティが表示されます。

2. **カテゴリ** アイコンまだ選択されていない場合は、これをクリックします。

プロパティが以下のメイン カテゴリにグループ分けされます。

結果 **テスト ステップ** ペインの列に対応するプロパティが含まれます。ステップの名前、ステップ実行の日付と時刻、ステップ実行に要した時間などのプロパティです。

拡張プロパティ Silk Test Workbench が、入力 操作ステップによって入力されたテキストのコントロールまたは値を識別するために使用するロケーター名など、追加の再生の詳細が含まれます。拡張プロパティは、操作ステップで変数や式が使用されたときにその内容を表示するのに便利です。たとえば、入力 ステップで変数 `textVar` が値として使用された場合、[拡張プロパティ] カテゴリに `textVar` の内容が表示されます。

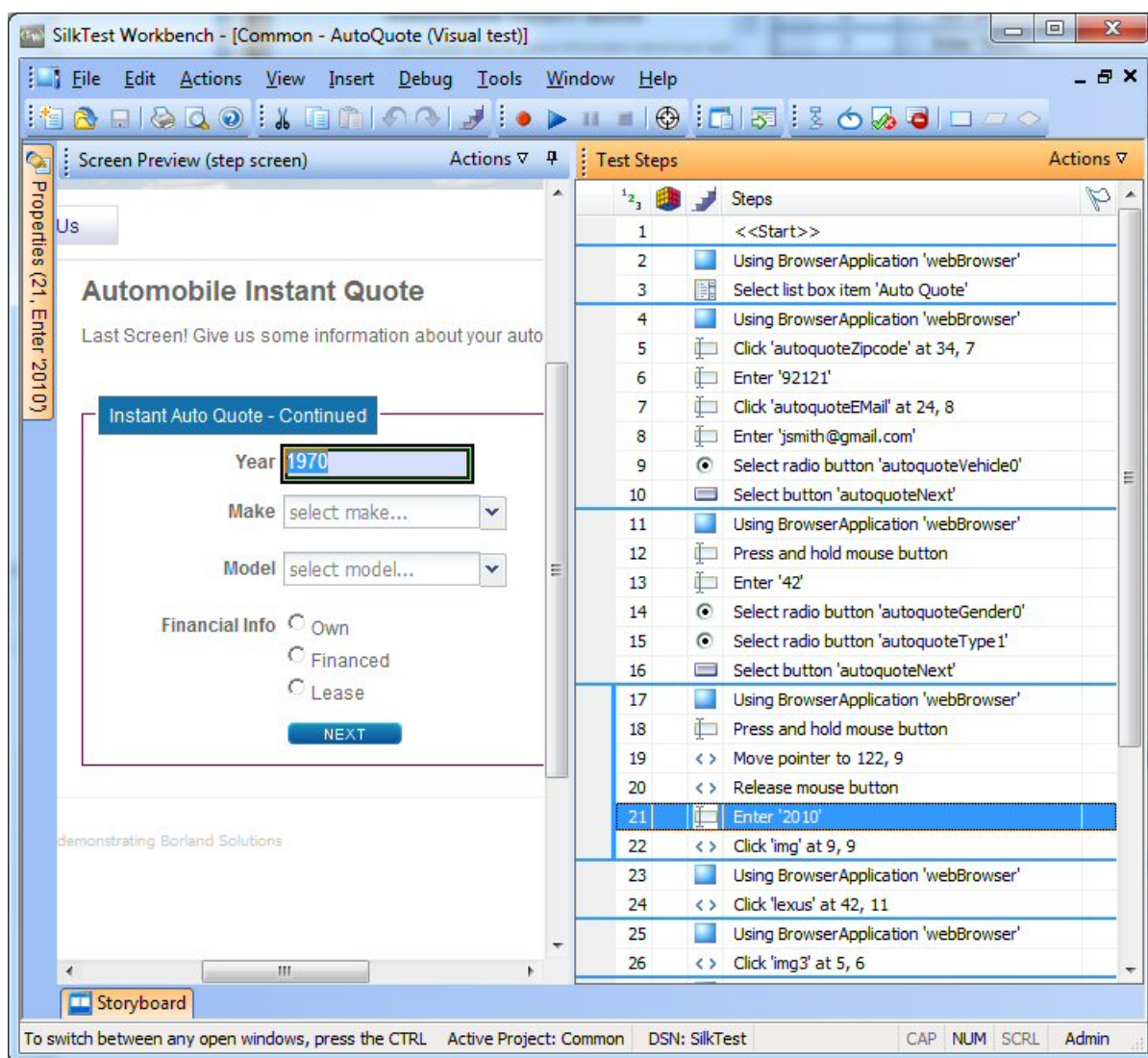
3. **拡張プロパティ** ノードを展開します。Text プロパティおよびロケーター プロパティがその値と共に示されます。Text プロパティの値は '42' です。これは、このステップで **Age** テキスト ボックスに入力された値です。

4. **再生前のビジュアル テストのステップ プロパティを表示/非表示** アイコンをクリックします。[ビジュアル テストの詳細]ノードが表示され、ビジュアル テストに存在するステップの元のプロパティがすべて表示されます。

画面プレビューの使用

画面プレビュー には、**テスト ステップ** ペインに表示された、コントロールと対話する各ステップについてキャプチャされたテスト アプリケーションのイメージが表示されます。キャプチャされたイメージとして、デスクトップ全体、アプリケーション ウィンドウ、またはアクティブなウィンドウのみを示すことができます。キャプチャされたイメージは、関連するステップが実行される前のアプリケーションの状態を示します。

1. **テスト ステップ** ペインで、車の製造年を含むステップを選択します。「2010」と入力します。**画面プレビュー** に、再生中にキャプチャされた画面とビジュアル テストの記録中にキャプチャされた画面が並んで表示されます。このステップのコントロールであるブラウザ ウィンドウが黒のボックスで強調表示されます。



2. 画面プレビューで、操作 > 相違点の表示 > オフ をクリックします。ビジュアルテストの画面が消え、再生の画面が開きます。
3. 操作 > ズーム > 75% をクリックしたあと、スクロールバーを使用して製造年がはっきりと表示されるまで移動します。
4. ビジュアルテストに切り替えます。テストステップペインで、操作 > ビジュアルテストのウィンドウ をクリックします。

ビジュアルテストの拡張：概要

ビジュアルテストの拡張には、既存のビジュアルテストが新しいバージョンのテストアプリケーションで使用できるようにテストを更新することなどが含まれます。たとえば、テストアプリケーションのさまざまな状態を処理し、検証するために、テストロジックを挿入することができます。さらに、ビジュアルテストの内容をわかりやすくしたり、重要な点について自分自身や他のユーザーに注意を促すために、フラグ、メッセージボックス、またはステップの説明を挿入することができます。

これらは、Silk Test Workbench で既存のビジュアルテストを拡張し、よりパワフルで堅牢かつ柔軟なテストを作成する、ほんの数例です。

画面プレビューからの更新

Silk Test Workbench はビジュアルテストを記録するとき、テストアプリケーションの各画面の関連するコントロールだけでなく、画面もキャプチャします。Silk Test Workbench により、画面プレビューには

キャプチャされた各画面が表示され、ビジュアルテストのステップによって識別されるコントロールが強調表示されます。**画面プレビュー**で、キャプチャされた画面の別のコントロールを識別することにより、テストアプリケーションにアクセスすることなくステップを更新できます。


このレッスンでは、Silk Test Workbench の **画面プレビューからのコントロールの挿入** 機能を使用して、**画面プレビュー** のキャプチャされた画面から別のボタンを選択します。

1. AutoQuote ビジュアルテストの **テストステップ** ペインで、<<終了>> ステップの前にあるステップを選択します。

ステップテキストは、次のようになります：(40, 5) の場所で 'Home' をクリックします

画面プレビュー に **Purchase A Quote** ページが表示され、**Home** ボタンが強調表示されます。

2. **画面プレビュー** で **操作 > ズーム > 75%** をクリックし、スクロールバーを使用して **Contact Us** ボタンがはっきり表示されるまで移動します。
3. **プロパティ** ペインで、**ロケータ** テキスト ボックスをクリックします。ロケータの値領域に、ロケータ選択ボタンが表示されます。
4. **プロパティ** ペインで、**画面プレビューから識別** をクリックします。

 **注:** ボタンが無効の場合、ビジュアルテストの記録を行う前にブラウザコントロールのキャプチャを有効にしていなかったことが原因です。

ポインタが **画面プレビュー** に移動します。

5. ポインタを **Contact Us** ボタンの上に移動し、**Contact Us** ボタンをクリックします。

ボタンをクリックすると、ボタンの新しいロケータ名を表示するように **プロパティ** ペインの **ロケータ** プロパティが変わり、**テストステップ** ペインのステップテキストが以下のように変わります。
(40, 5) の場所で 'Contact Us' をクリックします

6. **保存** をクリックします。

次にこのビジュアルテストを再生すると、Silk Test Workbench により **Home** ボタンではなく **Contact Us** ボタンがクリックされます。

検証の挿入

検証とは、ユーザー定義の条件を評価し、合格/失敗のメッセージや、オプションとしてフラグをビジュアルテストの再生結果に送信するテストロジックです。

このレッスンでは、検証を挿入して、見積もりで正しい車両モデルが使用されていることを確認します。

1. **表示 > テストステップ > ステップと画面** をクリックして、ステップと画面の両方が表示されていることを確認します。
2. 次のステップを選択します：「(91,11) の場所で 'rx400' をクリックします」
3. **画面プレビュー** で **操作 > ズーム > 75%** をクリックし、スクロールバーを使用して **Model** がはっきり表示されるまで移動します。
4. 以下のいずれかのステップを実行します。
 - ツールバーの **検証タイプのロジックの作成** をクリックします。
 - **挿入 > テストロジック > 検証** を選択します。

テストロジックデザイナー ウィザードの **ようこそ** ページが開きます。

5. **次へ** をクリックします。**ロジックの種類**の選択 ページが開きます。
6. **コントロールのプロパティ** をクリックし、**次へ** をクリックします。**プロパティに基づく条件の定義** ページが開きます。
7. **画面プレビューから識別** をクリックします。ポインタが **画面プレビュー** に移動します。

8. **Model** コンボ ボックスを選択します。**プロパティに基づく条件の定義** ページに、選択したコントロールの名前が、コントロールのプロパティと共に表示されます。コントロール名は、以下のように表示されます。

```
webBrowser.browserWindow.modelCombo
```

9. **プロパティの選択** グリッドで、**テキスト** にスクロールして選択します。

10 **条件の選択** が **等しい** に設定されていることを確認します。

11 **プロパティの選択** から Text プロパティを選択して、**期待値** テキスト ボックスに **RX400** が入力されていることを確認します。

12 **次へ** をクリックします。**検証の構築** ページが開きます。このページから、再生結果に送信する合格/失敗メッセージを定義できます。ページの上部に Silk Test Workbench が検証する条件が以下のように表示されます。

```
If "webBrowser.browserWindow.modelCombo"."Text" 等しい "RX400"
```

この条件は検証のロジックを定義しています。条件では、モデル タイプと選択されたタイプが比較されます。

13 デフォルトの合格の説明を The model type is correct に置き換え、デフォルトの失敗の説明を The model type is NOT correct に置き換えます。

14 **次へ** をクリックします。**要約** ページが開きます。

検証ステップは選択したステップのあとに挿入されます。検証ステップのテキストは以下のようになります。

```
「If "webBrowser.browserWindow.modelCombo"."Text" 等しい "RX400"」を検証します
```

15 検証の内容を確認したら、**完了** をクリックします。

サンプル アプリケーションのプロパティの値を検証するテスト ロジックを挿入することで、記録したビジュアル テストを拡張できました。

アプリケーション データを保存するローカル変数の作成

変数を使用すると、テストの別の場所や別のビジュアル テストまたはスクリプトで使用するデータ値を保存できるため、ビジュアル テストを拡張できます。データは他の種類のファイルに出力することもできます。

このレッスンでは、チュートリアルのアートのレッスンで使用できるように、電子メール アドレスを表示するコントロールのテキストを保存します。このためには、まずテキストを保存するローカル変数を作成する必要があります。

1. **テストステップ** ペインで、**操作 > 挿入 > 変数 > ローカルの追加** をクリックします。**ローカル変数の追加** ダイアログ ボックスが開きます。

2. **変数名** テキスト ボックスに strEmailAddress と入力します。

3. **型** リストから **テキスト** を選択します。

このレッスンでは、**初期値** テキスト ボックスを空のままにします。これは、後続のレッスンで変数に値を保存するためです。

テキスト 型では、変数値がテキスト、つまり文字列データ型として保存されます。

4. **OK** をクリックします。ビジュアル テストに対して新しい変数が保存されます。変数を作成したあとは、<<開始>> ステップからそれを参照し、定義を編集することができます。

5. **strEmailAddress** 変数を表示するには、**テストステップ** ペインで <<開始>> ステップを選択します。



ヒント: <<開始>> ステップは、すべてのビジュアル テストで常に最初のステップになります。

ステップのプロパティが **プロパティ** ペインに表示されます。<<開始>> ステップを選択している場合、**変数** カテゴリに **strEmailAddress** が **テキスト** 変数として表示されます。

各変数型の値領域は、現在定義されている各変数型の数を示します。このレッスンではテキスト変数を 1 つ作成したので、その種類の値領域には 1 項目がこのテストに関連付けられていることが示されています。

見積もりの電子メール アドレスを保存するローカル変数を作成したので、アプリケーションの電子メール アドレスを変数に保存します。

ローカル変数へのアプリケーション データの格納

サンプル アプリケーションの **Get Instant Auto Quote page** ページには、一意の電子メール アドレスが表示されます。このページの電子メール アドレスを含むテキストは、このページのコントロールのプロパティ値です。

このレッスンでは、前のレッスンで作成したローカル変数 *strEmailAddress* にこのテキストを保存します。

- 1. テスト ステップ** ペインで、電子メール アドレスの値を表示するステップを選択します。
ステップ テキストは次のようになります : 「jsmith@gmail.com」 を入力します
ステップを選択すると、**Get Instant Auto Quote page** ページのキャプチャされた画面が **画面プレビュー** に表示されます。
- 2. テスト ステップ** ペインで、**操作 > 挿入 > コントロールのプロパティ** をクリックします。
これにより、ビジュアル テストの選択したステップの直後にステップが挿入されます。ステップ テキストは次のようになります : 「コントロールの "" プロパティを取得します」
このステップのプロパティを編集して、**Get Instant Auto Quote** ページの電子メール アドレスのテキストをローカル変数に保存します。使用するコントロール、そのコントロールのプロパティ、およびプロパティの値を保存する変数を指定するように、このステップのプロパティを編集します。変数は、前のレッスンで作成した *strEmailAddress* です。
- 3. 画面プレビュー** で **操作 > ズーム > 50%** をクリックし、スクロールバーを使用して電子メール アドレスのテキストがはっきり表示されるまで移動します。
- 4. プロパティ** ペインで、**ローケター** テキスト ボックスをクリックします。ローケターの値領域に、ローケター選択ボタンが表示されます。
- 5. 画面プレビューから識別** をクリックします。ポインタが **画面プレビュー** に移動します。
- 6.** ポインタをページの **E-Mail** テキストの上に移動します。
強調表示のボックスがページのテキストを囲んでいることを確認し、強調表示された領域をクリックしてコントロールを識別します。
Silk Test Workbench により、**プロパティ** ペインの **ローケター** 情報が次の値に更新されます :
webBrowser.browserWindow.autoquoteEMail
- 7. プロパティ** ペインで **プロパティ** テキスト ボックスをクリックし、その値リストから **Text** を選択します。
- 8. (ローカル変数の選択...)** をクリックし、リストから **strEmailAddress** を選択します。
これで、コントロール、必要な値を含むコントロールのプロパティ、およびプロパティの値を保存する変数の指定が終了したので、ステップのテキストは次のようになるはず : 「コントロールの 'Text' プロパティを変数 'strEmailAddress' に格納します」
プロパティ ペインとステップ テキストの両方の **ローカル変数名** の値が *strEmailAddress* に変わります。

テストでプロパティ値が正しくキャプチャされ、保存されることを確認するために、テストを再生して結果を確認します。

拡張したビジュアル テストの再生と分析

記録したテストにいくつかの拡張を行ったので、ビジュアル テストを再生し、結果を分析してみます。

- 次のいずれか 1 つのステップを行います :
 - **操作 > 再生** を選択します。
 - ツールバーで **再生** をクリックします。

再生 ダイアログ ボックスが開きます。

2. **結果の説明** テキスト ボックスに、Enhanced test results for the recorded visual test と入力します。
3. **OK** をクリックします。
4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。Silk Test Workbench によって、拡張したテストが再生されます。
5. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。

結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。

要約 タブに、ビジュアル テストに合格した (エラーなしで正常に再生されたことを意味します)、または失敗した検証が表示されます。

6. **合格 (1)** タブをクリックします。
カッコ内の数は、合格した検証の合計数を示します。**テスト ステップ** ペインには検証ステップが表示され、**結果の詳細** 列には検証の合格の説明が表示されます。
7. **詳細** タブをクリックして、各ステップの結果を表示します。
8. **結果** ウィンドウの **テスト ステップ** ペインで、変数に電子メール アドレスを保存した結果を示すステップまでスクロールします。ステップ テキストは、次のようになります:「コントロールの 'Text' プロパティを変数 'strEmailAddress' に格納します」
9. *strEmailAddress* 変数の内容が **結果の詳細** 列に表示されます。**Text** プロパティの値全体を表示するには、カーソルをそのステップの **結果の詳細** 列の上に置きます。ツールヒントにプロパティの内容全体が表示されます。

お疲れ様でした。これで、サンプル アプリケーションを確実にテストするビジュアル テストを作成できました。次のレッスンでは、ビジュアル テストを他のビジュアル テストですばやく簡単に実行する方法など、より高度なテストの概念と機能について学習します。

ビジュアル テスト内でのビジュアル テストの実行 : 概要

このチュートリアルでは、Web アプリケーションから自動車保険の見積もりを取得するために必要な各操作を実行する単一のビジュアル テストを作成しました。単一のビジュアル テストは、単純なアプリケーションに対して基本的なテスト ケースを実行する場合に便利です。ただし、ほとんどのソフトウェア テストでは、アプリケーションのすべての側面をテストするなど、より厳格なアプローチが必要となります。さらに、テスト アプリケーションが変化したときに既存のビジュアル テストをすばやく更新することが必要です。

これらのテストの課題を解決する効率的な手段を提供するために、Silk Test Workbench はモジュール式テストをサポートしており、特定のテスト ソリューションの共通の操作を単一のテストにまとめ、同じ操作セットを必要とする他のビジュアル テストでビジュアル テストを再利用できるようにしています。

モジュール式テスト

ビジュアル テストやスクリプトなどの Silk Test Workbench 資産を作成してアプリケーションのテスト ソリューションを構築する前に、テスト戦略を立てることをお勧めします。

1 つのビジュアル テストまたはスクリプトに特定のテスト ソリューションのすべての部分を含める必要はありません。また、通常、そうすることは有益ではありません。

通常、最も効率のよいテスト方法は、モジュール式アプローチを採用したものです。アプリケーション テストを一連のトランザクション単位として考えます。

たとえば、オンライン発注システムのテストには、以下のようなトランザクション単位が含まれるかもしれません。

- オンライン システムへのログオン
- 顧客プロフィールの作成

- 注文の発注
- オンライン システムのログオフ

1 つのテストでこれらの単位をすべて処理し、このテストを使用するシナリオが 10 通りある場合、それらのシナリオを処理するために 10 個の別々のテストを記録する必要があります。アプリケーションに何らかの変更があった場合、たとえば、ログオン ウィンドウにフィールドが 1 つ追加された場合、新しいフィールドへのデータ入力を処理するために 10 個の別々のテストに変更が必要になります。

これらのトランザクション単位をすべてテストするビジュアル テストまたはスクリプトを 1 つ作成して、それをシナリオごとに 10 個作成するよりも、これらのトランザクション単位を 1 つずつ処理する別個のテストをテスト「モジュール」として作成する方が有益です。トランザクション単位ごとに別々のテストを作成し、それをテスト シナリオごとに再利用すれば、ログオン トランザクション単位を処理するテストだけを変更すればよいことになります。

モジュール式テストの基本が理解できたので、2 つめのテストを作成し、前のレッスンで作成したテストに追加してみます。

2 つめのビジュアル テストの記録

このセクションでは、2 つめのビジュアル テストを記録し、ビジュアル テスト資産の別の作成方法を学習します。

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **ビジュアル テスト** を選択し、**資産名** テキスト ボックスにビジュアル テストの名前を入力します。
このチュートリアルでは、名前として AddAccount と入力します。
3. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
4. **OK** をクリックして、ビジュアル テストを資産として保存し、記録を開始します。**テストの構成** ダイアログ ボックスが開きます。
5. **Web** タブを選択します。
6. リストから **Internet Explorer** を選択します。
7. サンプル アプリケーションの **Home** ページで、**Login** セクションの **Sign Up** をクリックします。**Create A New Account** ページが開きます。
8. 適切なフィールドに、以下の情報を入力します。
次のフィールドに移動するには Tab キーを押します。


フィールド名	値
名 (First Name)	Pat
姓 (Last Name)	Smith
生年月日 (Birthday)	12/02/1990
	 注: カレンダー コントロールの月と年の隣にある下矢印をクリックして月と年を変更し、カレンダーで 12 を選択します。
電子メール アドレス (E-mail Address)	smith@test.com
住所 (Mailing Address)	1212 Test Way
市 (City)	San Diego
州 (State)	CA
郵便番号 (Postal Code)	92121
パスワード (Password)	test

9. **Sign Up** をクリックします。
- 10 **Continue** をクリックします。連絡先情報が表示されます。
- 11 ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。
- 12 **Log Out** をクリックします。
- 13 **Alt+F10** を押して記録を完了します。**記録完了** ダイアログ ボックスが開きます。
- 14 **保存** をクリックします。ビジュアル ナビゲータにビジュアル テストが表示されます。


別のビジュアル テストへのビジュアル テストの挿入

このセクションでは、ユーザー アカウントを追加する 2 つめのビジュアル テストを、元のビジュアル テストの自動見積もりの要求を行うステップの前に挿入します。

ビジュアル テスト内でビジュアル テストを実行すると、複数のビジュアル テストで効率的に同じ基本ステップをテストできます。


 **ヒント:** ビジュアル テストを別のビジュアル テストに挿入する際、テスト アプリケーションが正しい初期再生状態になっていることを確認することが重要です。

1. **開始画面** の **最新** リストで、**AutoQuote** をクリックして開きます。
AutoQuote はこのチュートリアルで最初に作成したビジュアル テストです。
2. **テスト ステップ** ペインで <<開始>> ステップを選択します。
3. **操作 > 挿入 > ビジュアル テスト** をクリックします。**ビジュアル テストの参照** ダイアログ ボックスが開きます。
4. **資産の選択** リストから **AddAccount** という名前のビジュアル テストを選択し、**OK** をクリックします。

 **ヒント:** **資産の選択** リストに多くのビジュアル テストが表示されている場合、**名前フィルタ** テキスト フィールドに **AddAccount** を入力してから、ビジュアル テストをクリックすると簡単です。

Silk Test Workbench によって、選択したステップの前にステップが挿入されます。挿入されたステップによって、選択したビジュアル テストが呼び出されます。ステップ テキストは、次のようになります：

ビジュアル テスト 'AddAccount' の再生

 **ヒント:** 再生中に上記のステップが実行されると、挿入されたビジュアル テストが再生される前に、元のビジュアル テストが最後まで再生されます。

次のレッスンでは、このモジュール式ビジュアル テストを再生し、再生エラーに対処する方法を学習します。

再生エラーへの対応：概要

再生中のエラーは、テスト アプリケーションの変更、ビジュアル テストの不適切なステップ フローなど、さまざまな要因により発生します。このようなエラーをデバッグ機能を使用してすばやく診断し、修正することにより、テストの保守を最小限にし、チームによるテスト作業の効率を上げることができます。

まず、前のレッスンで作成したモジュール式テストを再生します。

モジュール式テストの再生

前のレッスンでは、ビジュアル テスト AddAccount を別のビジュアル テスト AutoQuote に挿入することにより、モジュール式テストを作成しました。

このセクションでは、このモジュール式テストを再生し、再生中のエラーを確認します。

1. AutoQuote ビジュアル テストが開いている状態で、ツールバーの **再生** をクリックします。**再生** ダイアログ ボックスが開きます。
2. **結果の説明** テキスト ボックスに Responding to errors in a modular test と入力します。

3. **OK** をクリックします。
4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。

再生中に、**Create A New Account** ページでテストが停止し、エラー メッセージが表示されます。データベースでは、各顧客レコードに対して一意の電子メール アドレスが必要なため、このエラーが発生します。AddAccount ビジュアル テストの記録中にすでに電子メール アドレスを入力しているため、この電子メール アドレスはデータベースにすでに存在しており、テストが失敗します。

再生エラーを確認したので、テストのデバッグを開始します。

エラーのデバッグ

Silk Test Workbench で再生エラーが発生すると、Silk Test Workbench の **再生エラー** ダイアログ ボックスが表示され、デバッグ モードを開始するオプションが提供されます。デバッグ モードでは再生が一時停止され、Silk Test Workbench のデバッグ機能を使用して再生エラーを診断し、修正することができます。

このセクションでは、前のセクションでモジュール式テストを再生しているときに発生したエラーをデバッグしてみます。

1. **再生エラー** ダイアログ ボックスで、**デバッグ** をクリックします。

デバッグ モードでは再生が一時停止されます。そのため、エラーを発生させたステップのプロパティを編集したり、ステップを削除または無効にしたり、他のビジュアル テストからステップをコピー アンド ペーストしたりしてエラーを修正してから、再生を再開できます。

Silk Test Workbench はデバッグ モードに入り、AddAccount ビジュアル テストを表示します。エラーを発生させたステップが黄色で強調表示されます。

2. **編集 > 有効化/無効化** を選択してステップ「'signup continue' をクリックします」を無効化します。

このステップを無効にすると、次回テストを再生するときにエラーを回避できます。

ステップのテキストがグレーの斜体になります。これはステップが無効にされたことを示します。

3. ツールバーで **再生** をクリックします。 **再生エラー** ダイアログ ボックスが表示されます。

4. **デバッグ** をクリックします。

5. **編集 > 有効化/無効化** を選択してステップ「'logout-form logout' をクリックします」を無効化します。

テストでは Home ページで **Log Out** ボタンを探しますが、Sign Up Continue ステップが無効になっておりボタンがページに表示されないために、このエラーが発生します。このエラーは、チュートリアルの方で修正します。

6. メニューから **デバッグ** を選択します。

メニューの上部に表示される以下のステップ コマンドを使って、ステップ実行を制御できます。

ステップ イン (F8) 1 つずつステップを再生します。**ステップ イン** は各ステップのトレースに有効であり、また再生しているビジュアル テストに挿入された別のビジュアル テストやスクリプトにもステップ インします。挿入された各ビジュアル テストやスクリプトについても、一度に 1 ステップまたは 1 行ずつ実行されます。

ステップ イン はテストの詳細分析に有効であり、変数の使用やターゲット アプリケーションとの対話に対する各ステップの影響を確認することができます。

ステップ オーバー (Shift +F8) 別のビジュアル テストに挿入されたビジュアル テスト全体またはスクリプト全体を、1 つのステップとして実行します。**ステップ オーバー** は、デバッグ モードの再生中に、ビジュアル テストまたはスクリプトを 1 ステップとして再生する場合に使用します。これにより、挿入されたビジュアル テスト全体またはスクリプト全体が再生されます。挿入されたビジュアル テスト全体またはスクリプト全体が再生されると、元のビジュアル テスト内の次のステップで再生が停止し、デバッグ モードになります。

別のビジュアル テストまたはスクリプトを再生するステップ以外のステップで **ステップ オーバー** を使用すると、**ステップ イン** を使用した場合と同じになります。次のステップのみが実行されたあと、再生は停止し、再びデバッグ モードになります。

ステップ アウト (Ctrl + Shift + F8) 別のビジュアル テストから再生しているビジュアル テストまたはスクリプト内の残りのステップをすべて実行したあと、元のビジュアル テスト内の次のステップで再生が一時停止します。

ステップ アウト は、別のビジュアル テストに挿入されたビジュアル テスト内のステップまたはコード行で再生が停止されているときに、残りのビジュアル テストを再生して元のビジュアル テストに戻る場合に使用します。挿入されたビジュアル テストまたはスクリプトの残りの部分の再生が実行されたら、元のビジュアル テスト内の次の行で再生が停止し、再びデバッグ モードになります。

カーソルまで実行 (Ctrl + F8) 再生を一時停止するステップを選択できます。これにより、ビジュアル テストまたはスクリプト内の選択したセクションを「ステップ オーバー」することができます。

カーソルまで実行 は、ビジュアル テストまたはスクリプトを再生して、実行時エラーが発生したポイントの直前で停止する場合に使用します。これにより、ブレークポイントを挿入することなく特定の行またはステートメントで再生を停止できます。再生が停止したら、別のデバッグ オプションのいずれかを使用して再生を継続できます。

カーソルから実行 現在選択されているテスト ステップからビジュアル テストを再生します。

7. ステップ アウト を選択します。

このコマンドにより、AddAccount ビジュアル テストの残りのステップが実行され、その後、AutoQuote ビジュアル テストの次のステップで再生が一時停止されます。

ステップ アウト を選択すると、Silk Test Workbench により AddAccount の最後のステップとしてホームページが表示され、再生が一時停止します。そして、Silk Test Workbench により AutoQuote がデバッグ モードで開かれ、次のステップが強調表示されます。

次に、ビジュアル テストで使用されている変数の値を監視する方法について学習します。

再生中の変数の追跡

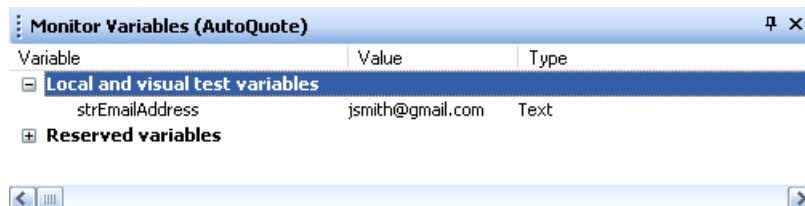
デバッグ モードでは、ローカル変数 ウィンドウを使用して、ローカル変数と予約変数の両方の再生値を追跡できます。

このセクションでは、ローカル変数 ウィンドウを使用して、前のレッスンで作成したローカル変数の値を参照します。

1. **デバッグ > ローカル変数** を選択します。ローカル変数 ウィンドウが開きます。

2. **ローカルおよびビジュアル テスト変数** ノードを展開します。

前のレッスンで作成したローカル変数 `strEmailAddress` が表示されます。再生中、この変数を使用するステップが実行されると、変数の値が **値** 列に表示されます。結果は、以下の図のようになります。



Variable	Value	Type
Local and visual test variables		
strEmailAddress	jsmith@gmail.com	Text
Reserved variables		



注: 変数の結果を ローカル変数 ウィンドウで確認する場合にテストでエラーがない場合は、ブレークポイントをビジュアル テストに挿入します。ブレークポイントを挿入するには、variable ス

トップのあとにあるステップを選択し、**デバッグ > ブレークポイントの設定/クリア** を選択してからテストを再生します。

3. **カーソルまで実行** デバッグの動作を確認するには、「(24, 9) の場所で 'Lexus' をクリックします」ステップを選択し、**デバッグ > カーソルまで実行** を選択します。Silk Test Workbench は、残りのステップを「(24, 9) の場所で 'Lexus' をクリックします」ステップまで再生したあと、AutoQuote ビジュアル テストを表示します。

次に、モジュール式テストの再生を完了し、テスト結果を確認します。

結果の確認

ビジュアル テストのデバッグが終了したら、ビジュアル テストの結果を確認します。

1. ツールバーの **再生** をクリックして、AutoQuote ビジュアル テストの再生を完了します。**再生完了** ダイアログ ボックスが開きます。
2. **結果へ移動** をクリックします。AutoQuote の結果が表示され、デフォルトで **要約** タブが表示されます。

要約 タブにはテスト実行の全体的な詳細が表示されます。**ビジュアル テストまたは .NET スクリプト (実行回数)** フィールドには、「AutoQuote(1)」と挿入されたビジュアル テスト「AddAccount(1)」がリストされます。

3. **詳細** タブをクリックします。
4. **テスト ステップ** ペインで、青色のテキストのステップまでスクロールします。

結果 列と **結果の詳細** 列を確認すると、再生中に発生したエラーに関する情報をすばやく参照できます。



注: テスト ステップ ペインにすべてのステップを表示するには、[ステップと画面] が選択されていることを確認します。**操作 > 表示 > ステップと画面** をクリックします。**失敗** タブには、再生エラーを含むステップは表示されません。失敗した検証のみが表示されます。

再生エラーを診断し、デバッグする方法を学びました。

アクティブ データの使用

アプリケーションの使用を効率よく模倣するために、アプリケーション テストでは異なるデータ セットを使用して同じ操作を繰り返すことがよく行われます。たとえば、前のレッスンでは顧客レコードを作成するテストを記録しました。10 件の顧客レコードを作成するには、テストごとに顧客データ セットを用意し、10 個の異なるテストを記録できます。ただし、Silk Test Workbench では元のテストを拡張し、実行ごとに異なるデータ セットを使用してテストを 10 回繰り返すことができます。

アクティブ データ テストでは、外部ファイルのデータをテスト アプリケーションへの入力として使用し、実行ごとに異なるデータを使用して、指定したステップを繰り返し実行することができます。

このレッスンでは、以下の操作方法を学びます。

- アクティブ データ テスト資産を作成し、ビジュアル テストに関連付ける
- 実行ごとに異なるデータを使用し、指定したステップを特定の回数実行する繰り返しロジックを作成する
- ビジュアル テストでアクティブ データ ファイルを使用する方法を定義する
- 繰り返し実行するステップを定義する
- データ ファイルのアクティブ データをビジュアル テストのリテラル データにマップする

前のレッスンでは、Pat Smith の顧客情報を顧客データベースに入力するビジュアル テストを記録しました。このビジュアル テストを再生するたびに、最初の記録中にキャプチャしたリテラル データ値が Silk Test Workbench で使用されます。

このレッスンでは、Pat Smith の顧客情報を入力するために使用されるリテラル データ値をアクティブ データに置き換え、Silk Test Workbench がビジュアル テストを再生するときに外部ファイルに含まれる別の顧客情報が使用されるようになります。

ビジュアルテスト用のアクティブ データを作成する前に、アクティブ データ ファイルについて確認します。

アクティブ データ ファイルの確認

アクティブ データ資産を作成する際、使用するデータが含まれる既存のデータ ファイルを選択するか、資産によって使用されるデータを入れる新しいファイルを作成することができます。このチュートリアルでは、customers.csv という名前の、顧客情報を含むカンマ区切りの既存のファイルを使用します。このファイルは、Silk Test Workbench のインストール ディレクトリの Examples フォルダにあります。

customers.csv ファイル内の各列は、Age 列を除き、データベースに顧客情報を入力するために使用されるフィールドに対応しています。Age 列は、このチュートリアルのあとのレッスンで使用します。

このテストにアクティブ データを使用すると、Silk Test Workbench はテストを再生し、ファイル内の各顧客がデータベースに入力されるまで、毎回別の顧客のデータを使って顧客情報を入力するステップを繰り返します。



ヒント: アクティブ データ ファイルでは、パスワード値はコード化されたテキストになっています。

アクティブ データ ファイルには以下の列が含まれます。

- 名 (First Name)
- 姓 (Last Name)
- 生年月日 (Birthday)
- 電子メール アドレス (Email Address)
- 住所 (Mailing Address)
- 市 (City)
- 州 (State)
- 郵便番号 (Postal Code)
- パスワード (Password)
- 年齢 (Age) (後続のレッスンで使用)

次に、アクティブ データ テスト資産を作成し、ビジュアルテストに関連付けます。

アクティブ データ テスト資産の作成

アクティブ データ テストでデータ ファイルを使用するには、そのファイルを使用するアクティブ データ資産を作成し、そのアクティブ データ資産をビジュアルテストに関連付ける必要があります。

1. 開始画面の **最新** リストで、**AddAccount** というビジュアルテストをダブルクリックして開きます。ビジュアルナビゲータに AddAccount ビジュアルテストが表示されます。
2. **テストステップ** ペインで、**操作 > 挿入 > アクティブ データ > 新規作成** をクリックします。**アクティブ データ資産の設定** ウィンドウが表示されます。
3. **名前** テキストボックスにアクティブ データ資産の名前を customers と入力します。
4. **参照** をクリックして、アクティブ データ資産と関連付ける customers.csv ファイルを探します。**アクティブ データ資産の選択** ダイアログボックスが表示されます。
5. サンプル ファイルの場所に移動し、アクティブ データのサンプル ファイル customers.csv を選択します。
デフォルトの場所は、次のようになります : C:\Program Files\Silk\Silk Test\examples\customers.csv
6. **開く** をクリックします。**全般** タブの **ファイル** テキストボックスにパスとファイル名が表示されます。
7. **アクティブ データ資産の設定** ウィンドウで、**オプション** タブをクリックします。
8. **最初の行を見出しとして使用** チェックボックスをオンにします。



ヒント: **詳細** タブをクリックすると、アクティブ データ ファイルの内容を確認することができます。

この設定により、アクティブ データ ファイルのデータの最初の行が、データではなく見出し行として扱われます。

9. **保存して閉じる** をクリックしてアクティブ データ資産を作成し、AddAccount ビジュアル テストに関連付けます。

ビジュアル テストでは、関連付けられているアクティブ データ資産の情報はすべて、<<開始>> ステップの **プロパティ** ペインに格納されます。ビジュアル テストに関連付けられている各アクティブ データ資産を確認するには、<<開始>> ステップを選択し、**プロパティ** ペインの **アクティブ データ** プロパティを確認します。

これで、アクティブ データ資産が作成され、ビジュアル テストと関連付けられました。次に、テストでアクティブ データを使用する繰り返しロジックを作成します。

アクティブ データ ファイルの繰り返しロジックの作成

ビジュアル テストでアクティブ データを使用する場合、通常は一連のステップを繰り返し、その際毎回リテラル データをアクティブ データに置き換えます。

このレッスンでは、データベースに顧客を追加するリテラル データを、10 件の顧客レコードを含むアクティブ データ ファイル customers.csv 内のデータに置き換えます。繰り返しごとに、アクティブ データ ファイルの顧客レコードのデータが使用されて、InsuranceCo Web サイトの **Create a New Account** ページの各フィールドにデータが入力されます。

1. ビジュアル ナビゲータに AddAccount ビジュアル テストが表示されていることを確認します。
2. Silk Test Workbench のツールバーの **繰り返しタイプのロジックの作成** をクリックします。 **ようこそ** ページが表示された **テスト ロジック デザイナ** ウィザードが表示されます。
3. **次へ** をクリックします。 **ロジックの種類を選択** ページが開きます。
4. **アクティブ データ ファイルを使用してステップ シーケンスを繰り返す** をクリックし、**次へ** をクリックします。 **使用するアクティブ データ資産の定義** ページが表示されます。
5. このビジュアル テストに関連付けられているアクティブ データ資産は 1 つだけ (customers) なので、**アクティブ データの資産** リストにこの資産が表示されます。そうでない場合は、リストからこの資産を選択します。
6. 省略可能： **シート名** リストからデータ ファイル中の使用するシートを選択します。デフォルトでは、アクティブ データ資産で指定されているシートが使用されます。
7. **開始行** テキスト ボックスで、デフォルト値の **1** をそのままにします。
8. **データを含む最後の行で終了** チェック ボックスをオンにします。
9. **順序通りにすべての行を取得** をクリックします。

これで、繰り返しロジックでデータ ファイルを使用する方法、および繰り返し回数が定義できました。

- 繰り返しロジックの実行時、アクティブ データ ファイルの最初の行が最初に使用されます。
- 各行を使って、最後の行まで繰り返されます。
- 繰り返しロジックではすべてのデータ行が順番に使用され、アクティブ データ ファイルにはデータ行が 10 行あるため、データベースに顧客を追加するステップは 10 回繰り返されます。毎回、アクティブ データ ファイルの異なる顧客データ行が入力として使用されます。

10 **次へ** をクリックします。 **繰り返しの構築** ページが開きます。

アクティブ データの使用方法が定義できたので、次にビジュアル テスト内の繰り返すステップを指定します。

繰り返しステップの定義

繰り返しロジックで繰り返すステップを決定するときは、リテラル データを置き換えるステップだけでなく、繰り返しが必要なすべての操作を考慮します。

このレッスンでは、リテラル データを置き換えるステップがいくつかありますが、アクティブ データ ファイル内の各顧客を入力するには、InsuranceCo Web サイトの **Create a New Account** ページにアクセス

するプロセス全体を繰り返す必要があります。そのため、ページにアクセスし **Home** ページにテストを戻すステップも含め、実際のテストプロセスのすべてのステップを繰り返す必要があります。

1. **繰り返しの構築** ページで **ステップの開始** リンクをクリックします。**ステップの選択** ダイアログ ボックスが開きます。
2. InsuranceCo Web サイトとの対話を開始するステップを選択します。これは <<開始>> ステップのすぐあとのステップです。
ビジュアル テストには顧客を追加するプロセスに必要なステップがすべて含まれるため、すべてのステップを繰り返す必要があります。
3. **OK** をクリックします。**ステップの選択** ダイアログ ボックスが閉じられ、選択したステップのテキストが **ステップの開始** リンクに表示されます。
4. **繰り返しの構築** ページで **ステップの終了** リンクをクリックします。**ステップの選択** ダイアログ ボックスが開きます。
5. InsuranceCo Web サイトとの対話を終了するステップを選択します。これは <<終了>> ステップのすぐ前のステップです。
6. **OK** をクリックします。**ステップの選択** ダイアログ ボックスが閉じられ、選択したステップのテキストが **ステップの終了** リンクに表示されます。
7. **次へ** をクリックします。**テスト ロジック デザイナー** の **要約** ページが表示されます。このページには、ビジュアル テストに定義されたテスト ロジックが表示されます。
8. **終了** をクリックして、ビジュアル テストのテスト ステップにテスト ロジックを挿入します。

Silk Test Workbench により、ビジュアル テストの <<開始>> ステップのあとに、繰り返しロジックを開始するステップが挿入されます。ステップのテキストは次のようになります:「アクティブ データ 'customers' を使用して繰り返します」

また、Silk Test Workbench により、<<終了>> ステップのあとに、繰り返しロジックを終了するステップも挿入されます。ステップのテキストは次のようになります:

繰り返しの終了

新しく挿入された繰り返しロジック ステップの間のすべてのステップがインデントされ、繰り返しロジック ステップの間にネストされて表示されます。ビジュアル テストでアクティブ データを使用するための最後の作業は、アクティブ データ ファイル内のデータをビジュアル テストのリテラル データにマップすることです。

アクティブ データのリテラル データへのマッピング

ビジュアル テストでアクティブ データ ファイル内のデータを使用するには、該当するテスト ステップのデータをアクティブ データ ファイル内のデータ列にマップする必要があります。

1. ビジュアル テストで、フィールドにデータを入力する最初のステップを選択します。
これは、ビジュアル テストで最初に記録された、InsuranceCo Web サイトの **Create a New Account** ページの **First Name** テキスト ボックスに「Pat」と入力するステップとなります。
ステップのプロパティが **プロパティ ペイン** に表示されます。
2. **プロパティ ペイン** の **パラメーター** カテゴリで、**text** パラメーターを選択します。これは、**Pat** に設定されたテキストを示します。
3. 値領域の **選択** をクリックし、**アクティブ データ** をクリックします。



ヒント: **customers** が **アクティブ データの資産** リストにない場合、この資産はこのビジュアル テストに関連付けられていません。

アクティブ データ列の選択 ダイアログ ボックスが開きます。このダイアログ ボックスで、テスト ステップの選択したプロパティのデータをアクティブ データ ファイル内の列のデータにマップします。

アクティブ データの資産 テキスト ボックスには、ビジュアル テストに関連付けられたアクティブ データ資産 **customers** が表示されます。**列** リストには、customers アクティブ データ ファイルの列名が表示されます。

4. 列 リストで、**First Name** 列を選択します。

これが名前を含んだファイル内の列であるため、この列をマップして、リテラル データ「Pat」を置き換えます。

5. **OK** をクリックしてダイアログ ボックスを閉じ、データをマップします。


Silk Test Workbench によって、テスト ステップの実際のデータが、アクティブ データ ファイル内の選択したデータへのマップ式に置き換えられます。つまり、既存のステップ テキスト Enter 'Pat' は、アクティブ データ資産名、置換するデータ型、および列名を表示するようになります。

```
Enter '[[customers].Text("First Name")]'
```

6. ビジュアル テストに関連付けられたアクティブ データ ファイル内のデータに置換するデータを含む残りのテスト ステップすべてについて、ステップ 1 から 5 を繰り返します。次のステップは、「'Smith' を入力します」というステップ テキストが含まれるステップです。

下の表は以下の内容を示します。

- このビジュアル テスト内の、アクティブ データ ファイルのデータで置き換えるリテラル データを含む各ステップのテキスト
- 置き換えるデータを含むステップのプロパティの名前
- リテラル データの代わりに使用するデータを含むアクティブ データの列

 **注:** アクティブ データを使用して更新された、パスワード保護テキスト ボックスにテキストを入力するステップについては、ステップのテキストにアクティブ データ情報が示されません。

ステップ テキスト	プロパティ	アクティブ データ列
'Smith' を入力します	テキスト	Last Name
'smith@test.com' を入力します	テキスト	Email Address
'1212 Test Way' を入力します	テキスト	Mailing Address
'San Diego' を入力します	テキスト	City
リスト ボックス項目 'California' を選択します	リスト ボックス項目	State
'92121' を入力します	テキスト	Postal Code
'test' を入力します	テキスト	Password

アクティブ データ ファイルのデータを使用するようにビジュアル テストを設定できたので、これを再生し、結果を確認します。

アクティブ データのビジュアル テストの再生と分析

顧客を追加するビジュアル テストを、アクティブ データ ファイルの顧客レコードを使用するように拡張できたので、再生してみます。このビジュアル テストを再生すると、アクティブ データ ファイルの顧客レコードが実際に顧客データベースに入力されます。

1. 次のいずれか 1 つのステップを行います：

- **操作 > 再生** を選択します。
- ツールバーで **再生** をクリックします。

再生 ダイアログ ボックスが開きます。

2. **結果の説明** テキスト ボックスに AddCustomers using ActiveData と入力します。

3. **OK** をクリックします。

4. 再生をサポートしている複数のブラウザがマシンにインストールされている場合、**ブラウザの選択** ダイアログ ボックスが開きます。ブラウザを選択して、**実行** をクリックします。Silk Test Workbench によって、拡張したテストが再生されます。


5. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。

結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。

6. 詳細 タブをクリックして、各ステップの結果を表示します。

詳細 タブには、再生中に実行された各ステップの結果が表示されます。スクロールバーを使用して、実行されたステップをスクロールします。

結果 ウィンドウの **テスト ステップ** ペインには、ビジュアル テストの実際のステップよりもかなり多くのステップがあります。これは、繰り返しごとに、それぞれ実行ステップのセットがあるためです。

 **注:** また、式を使用して、ビジュアル テストでアクティブ データを更新することもできます。**式デザイナ** を使用すると、アクティブ データ ファイルの列のデータを入力として使用でき、データの値を変更する式を作成できます。更新されたデータは、同じアクティブ データ ファイルの同じ列に保存できます。詳細については、オンライン ヘルプの「ビジュアル テストでの式を使用したアクティブ データの更新」を参照してください。

ビジュアル テストからのスクリプトの再生

スクリプトとビジュアル テストは、資産を使用してテスト アプリケーションでのメニュー項目の選択やデータの入力などの手動のユーザー操作を自動化するという点で同じです。違いは、ユーザーの操作がどのように表現されるかです。スクリプトでは、Microsoft .NET Framework で稼動するスクリプト記述言語である Microsoft の Visual Basic が使用されますが、ビジュアル テストでは、ビジュアル ナビゲータのポイントアンドクリック インターフェイスによって生成されるステップが使用されます。


いずれの場合も、互いに独立して実行できるパワフルで柔軟性の高い自動テストを作成できます。どちらを選ぶかは、ユーザーのニーズと好みによります。また、両方を組み合わせて使用することもできます。たとえば、特殊なタスクを実行するスクリプトを作成し、ビジュアル テストにそのスクリプトを再生するステップを挿入することができます。このようにして、スクリプトで使用されるスクリプト記述言語の力を利用してビジュアル テストを補強することができます。

ビジュアル テストからスクリプトを再生すると、スクリプトは、繰り返しテスト タスクまたは特殊なテスト タスクを実行するために必要に応じて呼び出せる関数と同様に機能します。このアプローチはチームによるテスト環境で役立ちます。たとえば、経験の豊富なテスト担当者が共通するテスト機能を実行するスクリプトのライブラリを作成し、経験の浅い開発者がそこからスクリプトを選択して、ビジュアル テストを作成することができます。

乱数を生成するスクリプトの作成

まず最初に、ユーザーの年齢に使用できる乱数を生成するスクリプトを作成します。

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスに `function_randomAge` と入力して **OK** をクリックします。

 **ヒント:** スクリプトに「function」と名前を付けることにより、共通して使用されるテスト タスクのライブラリを整理でき、ここからすばやくタスクを選択して、ビジュアル テストに挿入することができます。

スクリプトが **コード** ウィンドウに表示されます。

3. `Sub Main()` の下の行にカーソルを置き、以下のように入力します。

```
Dim rand As New Random()  
Dim TSrandomAge As Integer = rand.Next(10000, 99999)  
MsgBox(TSrandomAge)
```

4. ツールバーで **再生** をクリックします。**再生** ダイアログ ボックスが開きます。
5. **OK** をクリックします。メッセージ ボックスが表示され、10,000 から 99,999 の生成された乱数が表示されます。
6. **OK** をクリックします。**再生完了** ダイアログ ボックスが開きます。
7. **.NET スクリプトへ移動** をクリックします。

8. 数値範囲パラメータの (10000, 99999) をパラメータ *MinVal* と *MaxVal* に置き換えます。これら 2 つのパラメータを使用することにより、スクリプトを開いて値を変更するのではなく、このスクリプトを再生するビジュアル テストから乱数の範囲を設定することができます。以下のように入力します。

```
Dim TSrandomAge As Integer = rand.Next(MinVal, MaxVal)
```

9. スクリプトが適切に動作することを確認するために、Sub Main() の直後で、以下の値をパラメータに割り当てます。

```
Dim MinVal=16  
Dim MaxVal=105
```

スクリプトは以下のコードのようになっているはずですが。

```
Public Sub Main()  
    Dim rand As New Random()  
    Dim MinVal=16  
    Dim MaxVal=105  
    Dim TSrandomAge As Integer = rand.Next(MinVal, MaxVal)  
  
    MsgBox(TSrandomAge)  
End Sub
```

- 10 ツールバーで **再生** をクリックします。 **再生** ダイアログ ボックスが開きます。

- 11 **OK** をクリックします。メッセージ ボックスが表示され、16 から 105 の生成された乱数が表示されます。

- 12 **OK** をクリックします。 **再生完了** ダイアログ ボックスが開きます。

13. **NET スクリプトへ移動** をクリックします。

- 14 入力パラメータを使用して *MinVal* および *MaxVal* パラメータを設定するには、以下のステップを実行します。

- a) 作成する入力パラメータが含まれるように Main() sub を変更します。

```
Public Sub Main(args As IDictionary(Of String, Object))  
    Dim MinVal=args ("MinVal")  
    Dim MaxVal=args ("MaxVal")
```

"MinVal"と "MaxVal" は、次の手順で作成する入力パラメーターの名前です。

- b) 作成する出力パラメータが含まれるように Main() sub を変更します。

```
Public Sub Main(args As IDictionary(Of String, Object))  
  
    args ("TSrandomAge")=TSrandomAge
```

"TSrandomAge" は、次の手順で作成する出力パラメーターの名前です。

- c) 年齢はスクリプトからビジュアル テストに渡されるため、ステートメントの先頭にアポストロフィを入力してメッセージ ボックスのコードをコメントにします。

```
'MsgBox(TSrandomAge)
```

スクリプト全体は以下のようにになっているはずですが。

```
Public Module Main  
    Dim _desktop As Desktop = Agent.Desktop  
  
    Public Sub Main(args As IDictionary(Of String, Object))  
  
        Dim rand As New Random()  
  
        Dim MinVal= args("MinVal")  
        Dim MaxVal= args("MaxVal")  
  
        Dim TSrandomAge As Integer = rand.Next(MinVal, MaxVal)  
        args ("TSrandomAge")=TSrandomAge  
        'MsgBox(TSrandomAge)
```



```
End Sub
End Module
```

次に、ランダム年齢をビジュアル テストに渡すために使用するスクリプトの入力パラメータと出力パラメータを定義します。

スクリプトの入力パラメータの定義

スクリプトは、入力パラメータを使用してビジュアル テストからデータを受け取ったり、逆に出力パラメータを使用してビジュアル テストにデータを渡したりできます。このタスクでは、function_randomAge スクリプトで作成された乱数の範囲を設定する 2 つの入力パラメータを定義します。

1. **プロパティ ペイン**で、**入力パラメータの追加** を右クリックして選択します。**スクリプト入力パラメータの追加** ダイアログ ボックスが開きます。
2. **名前** テキスト ボックスに **MinVal** と入力します。
3. **型** リストで、**数値 (Double)** を選択します。
4. **デフォルト値** テキスト ボックスに 16 と入力します。
必要に応じて、ビジュアル テストにデフォルト値を設定することもできます。
5. **OK** をクリックします。**プロパティ ペイン**の入力パラメータのリストに、入力パラメータが表示されます。
6. **プロパティ ペイン**で、**入力パラメータの追加** を右クリックして選択します。**スクリプト入力パラメータの追加** ダイアログ ボックスが開きます。
7. **名前** テキスト ボックスに **MaxVal** と入力します。
8. **型** リストで、**数値 (Double)** を選択します。
9. **デフォルト値** テキスト ボックスに 105 と入力します。
必要に応じて、ビジュアル テストにデフォルト値を設定することもできます。
10. **OK** をクリックします。**プロパティ ペイン**の入力パラメータのリストに、入力パラメータが表示されます。

スクリプトの出力パラメータの定義

スクリプトは、入力パラメータを使用してビジュアル テストからデータを受け取ったり、逆に出力パラメータを使用してビジュアル テストにデータを渡したりできます。このタスクでは、function_randomAge スクリプトで作成された乱数をビジュアル テストに渡すパラメータを定義します。

1. **プロパティ ペイン**で、**出力パラメータの追加** を右クリックして選択します。**スクリプト出力パラメータの追加** ダイアログ ボックスが開きます。
2. **名前** テキスト ボックスに **TSrandomAge** と入力します。
3. **型** リストで、**数値 (Double)** を選択します。
4. **デフォルト値** テキスト ボックスは空のままにします。
5. **OK** をクリックします。**プロパティ ペイン**の出力パラメータのリストに、出力パラメータが表示されます。

ビジュアル テストで出力パラメータを受け取るローカル変数を定義します。

スクリプトのデータを使用するビジュアル テストの設定

スクリプトのデータを使用するようにビジュアル テストを設定するには、スクリプトを再生し、スクリプトのデータを格納するローカル変数を作成し、このローカル変数をスクリプトの出力変数と関連付けるステップを挿入する必要があります。さらに、ビジュアル テストからスクリプトにデータを渡すために、スクリプトの入力変数の値を設定する必要があります。

1. **AutoQuote** ビジュアル テストを開きます。

2. **ファイル > 名前を付けて保存** を選択します。 **名前を付けて保存** ダイアログ ボックスが開きます。
3. AutoQuote ビジュアル テストを変更するため、元のテストと区別できるよう **AutoQuote_Modified** と名前を変更し、**OK** をクリックします。
4. 以下のステップ テキストのあとにあるステップを選択します。
ビジュアル テスト 'AddAccount' の再生
5. **テスト ステップ** ペインで、**操作 > 挿入 > .NET スクリプト** を選択します。 **.NET スクリプトの参照** ダイアログ ボックスが開きます。
6. **function_randomAge** を選択し、**OK** をクリックします。 Silk Test Workbench により、スクリプトを再生するステップが挿入されます。
7. スクリプトのデータを格納するローカル変数を追加するには、以下のステップを実行します。
 - a) **テスト ステップ** ペインで、**操作 > 挿入 > 変数 > ローカルの追加** をクリックします。 **ローカル変数の追加** ダイアログ ボックスが開きます。
 - b) **変数名** テキスト ボックスに VTrandomAge と入力します。
 - c) **型** リストから **数値 (Double)** を選択し、**初期値** を 0 に設定します。
 - d) **OK** をクリックします。 <<開始>> ステップから変数を参照し、定義を編集することができます。
8. スクリプトのステップを選択します。
ステップは次のようになります : 「.NET スクリプト 'function_randomAge' の再生」
9. 挿入されたスクリプト ステップの **プロパティ** ペインで、**'TSrandomAge'** の内容の**引き渡し先** プロパティの値領域をクリックし、ローカル変数 **VTrandomAge** を選択します。
入力パラメータをスクリプトで追加したため、その値が **プロパティ** ペインに表示されます。ただし、値を編集することで、ビジュアル テストで使用される入力パラメータを変更できます。これらの値は、スクリプトでは変更されません。

これで、スクリプトのデータを使用するようにビジュアル テストが設定されました。

AutoQuote_Modified ビジュアル テストが再生されるたびに、function_randomAge スクリプトが再生され、ビジュアル テストのローカル変数 VTrandomAge に固有の乱数が渡されます。

次に、function_randomAge スクリプトで生成された乱数を含むビジュアル テストのローカルパラメータを使用するように、年齢情報を入力するステップを変更します。

ビジュアル テストでのスクリプト データの使用

このセクションでは、function_randomAge スクリプトで生成された乱数を含むビジュアル テストのローカルパラメータを使用するように、年齢情報を入力するステップを変更します。

1. **Age** テキスト ボックスで、データを入力するステップを選択します。ステップ テキストは、次のようになります :
'42' を入力します
2. **プロパティ** ペインで、**テキスト** プロパティを選択します。
3. **選択** ボタンをクリックして、**変数** を選択します。 **変数の選択** ダイアログ ボックスが開きます。
4. **VTrandomAge** を選択し、**OK** をクリックします。ステップ テキストは、次のようになります :
「'[VTrandomAge]' を入力します」

このステップで、VTrandomAge ローカル変数の値が **Age** テキスト ボックスに入力されるようになります。

テスト結果の再生と確認

1. 次のいずれか 1 つのステップを行います :

- **操作 > 再生** を選択します。
- ツールバーで **再生** をクリックします。

再生 ダイアログ ボックスが開きます。


2. **結果の説明** テキスト ボックスに Adding Customers using data passed from a script と入力します。
3. **OK** をクリックします。
4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。Silk Test Workbench によって、拡張されたテストが再生され、スクリプトから渡された乱数とビジュアル テストで定義した追加の顧客情報を使用して 10 人の新規顧客が作成されます。
5. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。
結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。
6. **詳細** タブをクリックして、各ステップの結果を表示します。


Silk Test Workbench スクリプト チュートリアル

Silk Test Workbench スクリプト チュートリアルによろこそ。このチュートリアルでは、スクリプトの作成、スクリプトの再生、および再生結果の分析に必要な基本ステップについて学習します。また、多数の機能を使用して、記録したスクリプトをすばやく更新および拡張する方法についても学習します。

このチュートリアルでは、Microsoft Visual Basic および Microsoft .NET Framework の基本的な知識があることが前提になっています。.NET Framework についての知識をお持ちでない場合は、Microsoft の Web サイトを参照してください。

このチュートリアルでは、Silk Test Workbench を使用して繰り返し可能なテストを作成する練習をするための Silk Test サンプル Web アプリケーション (<http://demo.borland.com/InsuranceWebExtJS/>) を使用します。

 **注:** このチュートリアルで使用されているサンプル アプリケーションは、Internet Explorer に合わせて設計され、最適化されています。チュートリアルのレッスンと同じユーザー体験を得るには、Microsoft Focus は Internet Explorer 以外のサポート対象ブラウザでチュートリアル サンプル アプリケーションを実行することをお勧めしません。

 **注:** Web アプリケーションを記録または再生する前に、システムにインストールされているすべてのブラウザ アドオンを無効にします。Internet Explorer でアドオンを無効にするには、**ツール > インターネット オプション** をクリックし、**プログラム** タブをクリックし、**アドオンの管理** をクリックし、アドオンを選択してから **無効にする** をクリックします。

各レッスンは前のレッスンの出力に基づいて進められるため、チュートリアルのレッスンは順番に完了してください。

スクリプトの記録 : 概要

サンプル Web アプリケーションで保険の見積もり要求を作成するための操作を行うと、Silk Test Workbench はその操作を記録します。スクリプトに必要な操作の記録を完了すると、記録されたスクリプトが **コード** ウィンドウに表示されます。

サンプル Web アプリケーションの起動

このチュートリアルでは、Silk Test サンプル Web アプリケーションを使用します。この Web アプリケーションは、デモ用に提供されています。

Internet Explorer で、Silk Test サンプル Web アプリケーションを使用します。チュートリアルのレッスンと同じユーザー体験を得るには、Internet Explorer 以外のサポート対象ブラウザでサンプル Web アプリケーションを実行しないでください。

1. テストの速度と信頼性を高めるために DOM 関数を記録するには、以下のステップを実行します。
 - a) **ツール > オプション** をクリックします。
 - b) **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。**記録** オプションが右側のパネルに表示されます。

- c) **xBrowser** をクリックします。
- d) **ネイティブなユーザー入力を記録する** リスト ボックスから、**いいえ** を選択します。
- e) **OK** をクリックします。



注: 通常、Web アプリケーションをテストする場合、DOM 関数ではなく、ユーザーの入力そのものを使用します。Flash および Java アプレットなどのプラグインや、AJAX を使用するアプリケーションは、ネイティブなユーザー入力ではサポートされますが、高レベルの API 記録ではサポートされません。

2. Web アプリケーションの記録または再生を行う前に、ブラウザのアドオンをすべて無効にする必要があります。ブラウザのすべてのアドオンを確実に無効にするには、以下のステップを実行します。
 - a) Internet Explorer で**ツール > インターネット オプション** を選択します。**インターネット オプション** ダイアログ ボックスが開きます。
 - b) **プログラム** タブをクリックし、**アドオンの管理** をクリックします。**アドオンの管理** ダイアログ ボックスが開きます。
 - c) アドオンのリストの**状態** 列で、各アドオンの状態が**無効** になっていることを確認します。**状態** 列に**有効** と表示されている場合は、アドオンを選択してから**無効にする** をクリックします。
 - d) **閉じる** をクリックし、**OK** をクリックします。
3. サンプル アプリケーションにリモートでアクセスするには、<http://demo.borland.com/InsuranceWebExtJS> をクリックします。サンプル アプリケーションの Web ページが開きます。

サンプル Web アプリケーションに対するスクリプトの記録

記録中は、記録を停止するまでテスト アプリケーションでのすべての操作 (Silk Test Workbench の操作は除く) が Silk Test Workbench によって記録されます。記録が完了したら、生成したスクリプトを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。
たとえば、タイトルに AutoQuote と入力します。
3. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
4. **OK** をクリックして、スクリプトを資産として保存し、記録を開始します。**アプリケーションの選択** ダイアログ ボックスが開きます。
5. **Web** タブを選択します。
6. リストから **Internet Explorer** を選択します。
7. **移動する URL の入力** テキスト ボックスに、demo.borland.com/InsuranceWebExtJS/ を入力します。
8. **OK** をクリックします。Silk Test Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。
9. Insurance Company Web サイトでは、次のステップのいずれかを行います：
 - a) **Select a Service or login** リスト ボックスから **Auto Quote** を選択します。**Automobile Instant Quote** ページが開きます。
 - b) 郵便番号と電子メール アドレスを適切なテキスト ボックスに入力し、自動車タイプをクリックして、**Next** をクリックします。
たとえば、郵便番号に 92121、電子メール アドレスに jsmith@gmail.com をそれぞれ入力し、自動車タイプとして Car を指定します。
 - c) 年齢を指定し、性別と運転履歴タイプをクリックして、**Next** をクリックします。
たとえば、年齢に 42 を入力し、性別と運転履歴タイプに Male および Good をそれぞれ指定します。
 - d) 製造年、車種、モデルを指定し、財務情報タイプをクリックして、**Next** をクリックします。
たとえば、製造年に 2010 と入力し、車種とモデルに Lexus および RX400 をそれぞれ指定し、財務情報タイプとして Lease を指定します。
指定した情報の概要が現れます。

e) **Purchase** をクリックします。

Purchase A Quote ページが開きます。

f) ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。

10Alt+F10 を押すか、**記録中** ウィンドウで **停止** をクリックするか、Silk Test Workbench のタスクバー アイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、スクリプトが表示されます。

11.NET スクリプトへ移動 をクリックします。スクリプトが **コード** ウィンドウに表示されます。

12保存 をクリックします。

記録されたスクリプトの確認

Silk Test Workbench では、それ以外のすべてのアプリケーション内の操作がすべて記録されます。指示に忠実に従っていれば、サンプル アプリケーションの Web サイトで実行した操作だけが Silk Test Workbench によってキャプチャされます。Silk Test Workbench は再生中にこれらの操作を繰り返します。

スクリプトは、以下のサンプルのようになります。

```
Imports SilkTest.Ntf.XBrowser
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        _desktop.Control("control1").TypeKeys("9")
        With _desktop.BrowserApplication("webBrowser")
            With .BrowserWindow("browserWindow")
                .DomTextField("autoquoteZipcode").SetText("92121")
                .DomTextField("autoquoteEMail").SetText("jsmith@gmail.com")
                .DomRadioButton("autoquoteVehicle0").Select()
                .DomButton("autoquoteNext").Select()
                .DomTextField("autoquoteAge").SetText("42")
                .DomRadioButton("autoquoteGender0").Select()
                .DomRadioButton("autoquoteType1").Select()
                .DomButton("autoquoteNext").Select()
                .DomTextField("autoquoteYear").SetText("2010")
                .DomElement("img").DomClick(MouseButton.Left, New Point(8, 9))
                .DomElement("lexus").DomClick(MouseButton.Left, New Point(87, 7))
                .DomElement("img3").DomClick(MouseButton.Left, New Point(11, 10))
                .DomElement("rX400").DomClick(MouseButton.Left, New Point(96, 11))
                .DomRadioButton("autoquoteFinInfo2").Select()
                .DomButton("autoquoteNext").Select()
                .DomLink("home").Select()
            End With
        End With
    End Sub
End Module
```

スクリプトが前述の例と完全に一致しないことがあります。各ユーザーは、それぞれ異なる方法でアプリケーションと対話するためです。たとえば、フォームに情報を入力するとき、各フィールドをクリックしていくユーザーもいれば、Tab キーを使用して移動していくユーザーもいます。結果は同じですが、これらの操作の Silk Test Workbench による記録内容は異なります。このような違いはありますが、スクリプトは正しく再生されるはずで

記録されたテストの再生

スクリプトを記録し、保存したら、再生して、スクリプトが適切に動作することを確認できます。

1. 次のいずれか 1 つのステップを行います：

- **操作 > 再生** を選択します。
- ツールバーで **再生** をクリックします。

再生 ダイアログ ボックスが開きます。このダイアログ ボックスで、結果をどのように保存するかを指定できます。

2. **結果の説明** テキスト ボックスに Initial test results for the recorded test と入力します。

3. **OK** をクリックします。

4. 再生をサポートしている複数のブラウザがマシンにインストールされている場合、**ブラウザの選択** ダイアログ ボックスが開きます。ブラウザを選択して、**実行** をクリックします。

各結果は、一意のテスト実行番号によって識別されます。

Silk Test Workbench が最小化され、スクリプトが再生されます。再生中、サンプル アプリケーションに対して、スクリプトの記録中に行った操作が画面上で再生されます。再生が正常に完了すると、**再生完了** ダイアログ ボックスが開きます。

5. **結果へ移動** をクリックします。**結果** ウィンドウが表示されます。

結果の分析：概要

テストを再生すると、Silk Test Workbench によってテスト結果が生成されます。テスト結果にはスクリプトの再生に関する情報が含まれています。スクリプトの名前、実行番号、各ステップが実行された日付と時刻、各ステップの合格/失敗のステータスなどの重要な情報が含まれます。

スクリプトの拡張：概要

テストの拡張には、既存のテストが新しいバージョンのテスト アプリケーションで使用できるようにテストを更新することなどが含まれます。たとえば、テスト アプリケーションのさまざまな状態を処理し、検証するために、テスト ロジックを挿入することができます。さらに、テストの内容をわかりやすくしたり、重要な点について自分自身や他のユーザーに注意を促すために、メッセージ ボックスを挿入することができます。

これらは、Silk Test Workbench で既存のテストを拡張し、よりパワフルで堅牢かつ柔軟なテストを作成する、ほんの数例です。

検証の挿入

検証とは、ユーザー定義の条件を評価し、合格/失敗のメッセージを再生結果に送信するテスト ロジックです。

このレッスンでは、検証を挿入して、見積もりで正しい車両モデルが使用されていることを確認します。

1. 自動見積もりのモデル タイプを定義する、以下のテキストを選択します。

```
.DomElement("RX400").DomClick(MouseButton.Left, New Point(96, 11))
```

2. モデル (**Model**) タイプを指定する Instant Quote ウィザードのページに移動し、異なるモデル タイプに注目します。

たとえば、**GS430** は車種 **Lexus** のモデル タイプです。

3. Silk Test Workbench でモデル タイプを変更します。

textContents コードを

```
.DomElement("RX400").DomClick(MouseButton.Left, New Point(96, 11))
```

から次のように変更します：

```
.DomElement("GS430").DomClick(MouseButton.Left, New Point(96, 11))
```

4. 期待値と実際の値を比較して、コメントを追加するには、以下のように入力します。

```
Workbench.Verify ("GS430", .DomTextField("modelCombo").Text, "The model type is correct")
```

サンプルアプリケーションのプロパティの値を検証するテストロジックを挿入することで、記録したスクリプトを拡張できました。

アプリケーションデータの作成とローカル変数への保存

変数を使用すると、スクリプトの別の場所で使用するデータ値を保存できるため、テストを拡張できます。データは他の種類のファイルに出力することもできます。

サンプルアプリケーションの **Get Instant Auto Quote** ページには、一意の電子メールアドレスが表示されます。このページの電子メールアドレスを含むテキストは、このページのコントロールのプロパティ値です。

このレッスンでは、ローカル変数 *stremailAddr* に、このテキストを保存します。

1. スクリプトで、電子メールの値に移動します。

コードは以下のようになります。

```
.DomTextField("autoquoteEMail").SetText("jsmith@gmail.com")
```

2. 電子メールの値のあとに、以下のコードを挿入します。

```
Dim StremailAddr As String  
StremailAddr = .DomTextField("autoquoteEMail").Text
```

このステップによって、新しいローカル変数 *StremailAddr* が作成されます。このローカル変数に、**Get Instant Auto Quote page** ページの電子メールアドレスのテキストを保存します。

3. 再生中に変数テキストを表示する出力を含めるには、スクリプトに `Console.Write` コマンドを含めます。以下に例を示します。

```
Console.Write (StremailAddr)
```

4. コンソール出力を表示するには、**表示 > 出力** を選択します。**出力** ウィンドウが開きます。スクリプトを再生すると、**出力** ウィンドウが入力されます。

テストでプロパティ値が正しくキャプチャされ、保存されることを確認するために、テストを再生して結果を確認します。

拡張したスクリプトの再生と分析

記録したスクリプトにいくつかの拡張を行ったので、スクリプトを再生し、結果を分析してみます。

1. 次のいずれか 1 つのステップを行います：

- **操作 > 再生** を選択します。
- ツールバーで **再生** をクリックします。

再生 ダイアログ ボックスが開きます。

2. **結果の説明** テキスト ボックスに、Enhanced test results for the script と入力します。
3. **OK** をクリックします。

4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。Silk Test Workbench により、拡張されたテストが再生されます。

5. **再生完了** ダイアログ ボックスで **結果へ移動** をクリックします。

結果 ウィンドウが開き、デフォルトで **要約** タブが表示されます。

要約 タブに、スクリプトが合格した（エラーなしで正常に再生されたことを意味します）、または失敗した検証が示されます。

6. **合格 (1)** タブをクリックします。

カッコ内の数は、合格した検証の合計数を示します。**テストステップ** ペインには検証ステップが表示され、**結果の詳細** 列には検証の合格の説明が表示されます。

7. **詳細** タブをクリックすると、各操作の結果を表示されます。
8. **出力** ウィンドウで、変数に電子メール アドレスを保存した結果を示すステップまでスクロールします。
テキストは以下のようなものです。

```
jsmith@gmail.com
```

9. **合格** タブをクリックすると、モデル タイプの検証結果が表示されます。
テキストは以下のようなものです。

```
Main:Verify 合格:The model type is correct
```

検証結果は、**結果** 列および **結果の詳細** 列にも表示されます。

お疲れ様でした。これで、サンプル アプリケーションを確実にテストするスクリプトを作成できました。次のレッスンでは、スクリプトを他のスクリプトですばやく簡単に実行する方法など、より高度なテストの概念と機能について学習します。

スクリプト内でのスクリプトの実行：概要

このチュートリアルでは、Web アプリケーションから自動車保険の見積もりを取得するために必要な各操作を実行する単一のスクリプトを作成しました。単一のスクリプトは、単純なアプリケーションに対して基本的なテスト ケースを実行する場合に便利です。ただし、ほとんどのソフトウェア テストでは、アプリケーションのすべての側面をテストするなど、より厳格なアプローチが必要となります。さらに、テストアプリケーションが変化したときに既存のスクリプトをすばやく更新できることが必要です。

これらのテストの課題を解決する効率的な手段を提供するために、Silk Test Workbench はモジュール式テストをサポートしており、特定のテスト ソリューションの共通の操作を単一のテストにまとめ、同じ操作セットを必要とする他のスクリプトでスクリプトを再利用できるようにしています。

モジュール式テスト

ビジュアル テストやスクリプトなどの Silk Test Workbench 資産を作成してアプリケーションのテスト ソリューションを構築する前に、テスト戦略を立てることをお勧めします。

1 つのビジュアル テストまたはスクリプトに特定のテスト ソリューションのすべての部分を含める必要はありません。また、通常、そうすることは有益ではありません。

通常、最も効率のよいテスト方法は、モジュール式アプローチを採用したものです。アプリケーション テストを一連のトランザクション単位として考えます。

たとえば、オンライン発注システムのテストには、以下のようなトランザクション単位が含まれるかもしれません。

- オンライン システムへのログオン
- 顧客プロファイルの作成
- 注文の発注
- オンライン システムのログオフ

1 つのテストでこれらの単位をすべて処理し、このテストを使用するシナリオが 10 通りある場合、それらのシナリオを処理するために 10 個の別々のテストを記録する必要があります。アプリケーションに何らかの変更があった場合、たとえば、ログオン ウィンドウにフィールドが 1 つ追加された場合、新しいフィールドへのデータ入力を処理するために 10 個の別々のテストに変更が必要になります。

これらのトランザクション単位をすべてテストするビジュアル テストまたはスクリプトを 1 つ作成して、それをシナリオごとに 10 個作成するよりも、これらのトランザクション単位を 1 つずつ処理する別個のテストをテスト「モジュール」として作成する方が有益です。トランザクション単位ごとに別々のテストを作成し、それをテストシナリオごとに再利用すれば、ログオン トランザクション単位を処理するテストだけを変更すればよいことになります。

モジュール式テストの基本が理解できたので、2 つめのテストを作成し、前のレッスンで作成したテストに追加してみます。

2 つめのスクリプトの記録

このセクションでは、チュートリアル の 2 つめのスクリプトを記録し、スクリプトの別の作成方法を学習します。

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。
このチュートリアルでは、名前として AddAccount と入力します。
3. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
4. **OK** をクリックして、スクリプトを資産として保存し、記録を開始します。**アプリケーションの選択** ダイアログ ボックスが開きます。
5. **Web** タブを選択します。
6. リストから **Internet Explorer** を選択します。
7. サンプル アプリケーションの **Home** ページで、**Login** セクションの **Sign Up** をクリックします。**Create A New Account** ページが開きます。
8. 適切なフィールドに、以下の情報を入力します。
次のフィールドに移動するには Tab キーを押します。


フィールド名	値
名 (First Name)	Pat
姓 (Last Name)	Smith
生年月日 (Birthday)	12/02/1990
	 注: カレンダー コントロールの月と年の隣にある下矢印をクリックして月と年を変更し、カレンダーで 12 を選択します。
電子メール アドレス (E-mail Address)	smith@test.com
住所 (Mailing Address)	1212 Test Way
市 (City)	San Diego
州 (State)	CA
郵便番号 (Postal Code)	92121
パスワード (Password)	test

9. **Sign Up** をクリックします。
10. **Continue** をクリックします。連絡先情報が表示されます。
11. ページ上部にある **Home** をクリックして、記録を開始したホーム ページに戻ります。
12. **Log Out** をクリックします。
13. Alt+F10 を押して記録を完了します。**記録完了** ダイアログ ボックスが開きます。
14. **保存** をクリックします。スクリプトが **コード** ウィンドウに表示されます。

別のスクリプトへのスクリプトの挿入

このセクションでは、ユーザー アカウントを追加する 2 つめのスクリプトを、元のスクリプトの自動見積もりの要求を行うコードの前に挿入します。

スクリプト内でスクリプトを実行すると、スクリプトで同じ基本操作を効率的にテストできます。

 **ヒント:** スクリプトを別のスクリプトに挿入する際、テストアプリケーションが正しい初期再生状態になっていることを確認することが重要です。

1. **ファイル > 開く** を選択します。 **アセットブラウザ** が開きます。
2. 左側のペインで **.NET スクリプト** を選択し、スクリプトのリストを表示します。
3. リストから **AutoQuote** をダブルクリックして開きます。
AutoQuote はこのチュートリアルで最初に作成したテストです。
4. **コード** ウィンドウの `Public Sub Main()` コードの後ろにカーソルを置き、**Enter** を押して、新規行を追加し、以下のように入力します。

```
Workbench.RunScript ("AddAccount")
```

ここで、*AddAccount* は、作成した 2 つめのスクリプトの名前です。

見積もりステップを実行する前にアカウント情報を追加するため、**With** ステートメントの前に `Workbench.RunScript` コマンドが追加してあります。見積もりステップのあとで *AddAccount* スクリプトを実行するには、`Workbench.RunScript` コマンドを **End With** ステートメントのあとに追加します。

再生エラーへの対応：概要

再生中のエラーは、テストアプリケーションの変更、不適切なワークフローなど、さまざまな要因により発生します。このようなエラーをすばやく診断し、修正することにより、テストの保守を最小限にし、チームによるテスト作業の効率を上げることができます。

まず、前のレッスンで作成したモジュール式スクリプトを再生します。

モジュール式スクリプトの再生

前のレッスンでは、*AddAccount* を *AutoQuote* スクリプト挿入することにより、モジュール式スクリプトを作成しました。

このセクションでは、このモジュール式スクリプトを再生し、再生中のエラーを確認します。

1. *AutoQuote* スクリプトが開いている状態で、ツールバーの **再生** をクリックします。 **再生** ダイアログボックスが開きます。
2. **結果の説明** テキスト ボックスに `Responding to errors in a modular test` と入力します。
3. **OK** をクリックします。
4. 再生をサポートしている複数のブラウザーがマシンにインストールされている場合、**ブラウザーの選択** ダイアログボックスが開きます。ブラウザーを選択して、**実行** をクリックします。

再生中に、**Create A New Account** ページでテストが停止し、エラーメッセージが表示されます。

データベースでは、各顧客レコードに対して一意の電子メール アドレスが必要なため、このエラーが発生します。*AddAccount* スクリプトの記録中にすでに電子メール アドレスを入力しているため、この電子メール アドレスはデータベースにすでに存在しており、テストが失敗します。

結果の確認

スクリプトの結果を確認します。

1. **終了** をクリックして、再生を停止します。 **再生完了** ダイアログボックスが開きます。
2. **結果へ移動** をクリックします。 *AutoQuote* の結果が表示され、デフォルトで **要約** タブが表示されます。

要約 タブにはテスト実行の全体的な詳細が表示されます。 **ビジュアル テストまたは .NET スクリプト (実行回数)** フィールドには、「*AutoQuote*(1)」と挿入されたスクリプト「*AddAccount*(1)」がリストされます。

3. **詳細** タブをクリックします。

4. 青色のテキストのステップまでスクロールします。

結果 列と **結果の詳細** 列を確認すると、再生中に発生したエラーに関する情報をすばやく参照できます。



注: 失敗 タブには、再生エラーを含むステップは表示されません。失敗した検証のみが表示されます。

再生エラーを診断する方法を学びました。このチュートリアル目的のため、一旦スクリプトの再生を成功するためにスクリプトの電子メールを手動で変更してエラーを回避することができます。つまり、`AddAccount` スクリプトの

```
smith@test.com
```

を次のように変更します：

```
psmith@test.com
```

ヘルプについてのヘルプ

このセクションには、以下の情報があります。

- Silk Test Workbench ヘルプ
- ヘルプで使用されている表記規則

Silk Test Workbench ヘルプ

Silk Test Workbench のヘルプには、概念的な概要と手順を説明するトピックがあります。これらのトピックのタイプでは、必要に応じて、一般的な情報から特定の情報に移動できます。

概念的な概要 概念的な概要では、Silk Test で作業するために必要な製品のアーキテクチャ、コンポーネント、およびその他の情報を提供します。ほとんどの概要の最後には、関連するより詳細な情報へのリンクがあります。

手順の説明 手順の説明では、ステップごとに説明を提供します。ほとんどの手順の最後には、関連する手順へのリンクがあります。また、ほとんどの概念的な概要には、関連する手順へのリンクがあります。

ヘルプで使用されている表記規則

Silk Test Workbench のヘルプでは、以下の表記規則が使用されます。

規則	表す内容
等幅フォント	入力する必要があるソース コードおよびテキスト。
ボールド体	ダイアログ ボックスおよびその他のユーザー インターフェイス要素への参照。
イタリック体	変数などの識別子。イタリック体のテキストは新しい用語を強調するためにも使用されます。

ヘルプを表示する方法

Silk Test Workbench ヘルプ メニューには以下のコマンドがあります。

- **目次** をクリックすると、Silk Test Workbench ヘルプの目次が表示されます。
- **検索** をクリックすると、フルテキスト検索を使用してトピックを検索できます。
- **索引** をクリックすると、キーワードの索引を使用してトピックを検索できます。

- **チュートリアル** をクリックすると、Silk Test Workbench のチュートリアルを実行できます。このチュートリアルでは、サンプル アプリケーションでビジュアル テストを実際に作成、実行しながら、Silk Test Workbench の機能を学べます。
- Silk Test Workbench の**バージョン情報** をクリックすると、現在インストールされているバージョン、著作権情報、およびシステム情報が表示されます。

選択したダイアログ ボックスにある **ヘルプ** ボタンをクリックすることによっても、Silk Test Workbench ヘルプにアクセスできます。

ヘルプ トピックの印刷

1. **目次** タブをクリックします。
目次 タブが表示されていない場合は、**表示** をクリックして、ナビゲーション ペインを表示します。
2. 印刷するトピックを選択します。
3. **印刷** をクリックします。
4. 複数のトピックを印刷するには、ブックを選択し、**印刷** をクリックして、**選択された見出しおよびすべてのサブトピックを印刷** オプション ボタンをクリックします。
トピックは見出しごとに別のページに印刷されるのではなく、連続して印刷されます。

使用状況データの収集の有効化/無効化

Micro Focus では、テスト全体にわたるユーザー エクスペリエンスの向上を図るため、Micro Focus ソフトウェアとサービスの使用方法に関する情報を収集し、Micro Focus に提供していただくことを望んでいます。Silk Test Workbench のインストール時に使用許諾契約の条項に同意することにより、Silk Test Workbench の使用方法に関する情報と、Silk Test Workbench をインストールしたコンピューターに関する情報の収集を Micro Focus に許可したことになります。Micro Focus は、名前や住所などの個人の特定を可能にする情報の収集や、スクリプトやパスワードなどのデータ ファイルの収集は行いません。この情報の収集を Micro Focus に許可することにより、Micro Focus では、その傾向と使用方法のパターンの識別に役立てることができま。

Micro Focus による使用状況データの収集を有効化または無効化するには：

1. メニューから **へるぷ > Silk Test Workbench のバージョン情報** をクリックします。
2. バージョン情報ダイアログ ボックスで、**カスタム フィードバック オプション** をクリックします。
3. 以下のオプションのいずれかを選択します。
 - 使用状況データの収集を有効化するには、**はい、参加します** をクリックします。
 - 使用状況データの収集を無効化するには、**いいえ、参加しません** をクリックします。
4. **OK** をクリックします。

Micro Focus へのお問い合わせ

Micro Focus は、世界的規模のテクニカル サポートおよびコンサルティング サービスを提供します。すべての顧客のビジネスを成功に導くために、信頼できるサービスをタイムリーに提供するように、Micro Focus はワールドワイドのサポート体制を整えています。

保守およびサポート契約を結んだすべてのお客様、および製品を評価中のお客様は、カスタマー サポートを受けることができます。高度なトレーニングを積んだスタッフが、お客様の質問にできる限り迅速かつ専門的にお答えします。

<http://supportline.microfocus.com/assistedservices.asp> にアクセスするか、またはメールを supportline@microfocus.com に送信して、Micro Focus SupportLine と直接連絡できます。

また、<http://supportline.microfocus.com> の Micro Focus SupportLine では、最新のサポートに関するニュースや、さまざまなサポート情報を得ることができます。このサイトに初めてアクセスした場合は、ユーザー登録が必要な場合があります。

Micro Focus SupportLine が必要とする情報

Micro Focus SupportLine をご利用の場合は、可能な限り次の情報を提供ください。情報が多ければ多いほど、Micro Focus SupportLine はお客様に適切なサービスを提供できます。

- 問題の原因と思われるすべての製品の名前およびバージョン番号
- 使用しているコンピュータの製造元およびモデル
- システム情報 (オペレーティング システムの名前やバージョン、プロセッサやメモリの詳細など)
- 問題の詳細な説明 (問題の再現手順など)
- 発生したエラー メッセージ
- お客様のシリアル番号

これらの番号は、Micro Focus から受け取った Electronic Product Delivery Notice 電子メールの件名および本文に記述されています。

テストの作成

ビジュアルテストとスクリプトを作成する場合のプロセスおよび考慮事項について説明します。

ビジュアルテストの作成

ビジュアルテストは、一連の単純な操作が記録されたものである場合もあれば、関数呼び出しやアクティブデータが含まれる複雑なものである場合もあります。

ビジュアルテストをすばやく作成するには、記録機能を使用するのが最適の方法です。記録機能により、Silk Test Workbench が生成するビジュアルテストのテストステップを学習することができます。これにより、ビジュアルテスト作成の基本を理解し、効率的なテストソリューションの作成についての理解を深めることができます。

記録中、Silk Test Workbench との対話を除くすべての操作が記録されます。ただし、記録を中断して、検証を挿入することもできます。最初の記録が終了したら、ビジュアルテストを変更できます。

ビジュアルテストの記録

記録中は、記録を停止するまで、テストアプリケーションでのすべての操作（Silk Test Workbench の操作は除く）が Silk Test Workbench によって記録されます。記録が完了したら、生成したビジュアルテストを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. 省略可能 : Web アプリケーションに対してテストをする際に、画面プレビューから記録したコントロールを更新できるようにするには、ビジュアルテストを記録する前にブラウザコントロールのキャプチャを有効化してください。

詳細については、「記録の出力オプションの設定」を参照してください。



注: Web アプリケーションの記録中にコントロールのキャプチャを有効化すると、ビジュアルテストのパフォーマンスが低下する恐れがあります。Micro Focus では、コントロールのキャプチャを無効化することをお勧めします。

2. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログボックスが開きます。
3. [資産の種類] リストから **ビジュアルテスト** を選択し、**資産名** テキストボックスにビジュアルテストの名前を入力します。



注: デフォルトでは、アクティブになっているプロジェクトにビジュアルテスト資産が作成されます。別のプロジェクトにビジュアルテスト資産を作成するには、**プロジェクト** リストで該当するプロジェクトを選択します。

4. 省略可能 : Silk Test Workbench の記録オプションを変更する場合は、**オプション** をクリックします。
5. 記録をすぐに開始するために、**記録の開始** チェックボックスをオンにします。
6. **OK** をクリックして、ビジュアルテストを資産として保存し、記録を開始します。

記録の開始 チェックボックスをオンにせずに **OK** をクリックすると、Silk Test Workbench によりビジュアルテストが資産として保存され、**ビジュアルナビゲータ** が表示されます。

アプリケーションの選択 ダイアログボックスが開きます。



注: テストするアプリケーションがリストに表示されない場合は、**キャプションを持たないプロセスを表示しない** チェックボックスをオフにします。このオプションはデフォルトでオンになっており、キャプションが付いているアプリケーションのみをフィルタ処理するために使用します。

7. 現在のプロジェクトに対してアプリケーション構成が設定されていない場合、テストするアプリケーションの種類に対応するタブを選択します。

- ブラウザで実行しない標準アプリケーションをテストする場合は、**Windows** タブを選択します。
 - Web アプリケーションまたはモバイル Web アプリケーションをテストする場合は、**Web** タブを選択します。
 - ネイティブ モバイル アプリケーションをテストする場合は、**モバイル** タブを選択します。
8. 標準アプリケーションをテストするには、リストからアプリケーションを選択します。
9. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。
- a) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザのインスタンスが既に実行されている場合、**実行中のブラウザの URL を使用する** をクリックして、実行中のブラウザ インスタンスに現在表示されている URL の記録を行うことができます。
 - b) 省略可能：あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザ ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
10. ネイティブ モバイル アプリケーション (アプリ) をテストするには：
- a) アプリをテストするモバイル デバイスをリストから選択します。
 - b) ネイティブ モバイル アプリケーションを選択します。
 - モバイル デバイスまたはエミュレータ上にアプリをインストールする場合は、**参照** をクリックしてアプリ ファイルを選択するか、**アプリ ファイル** テキスト フィールドにアプリ ファイルへの完全パスを入力します。このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
 - Android デバイス上に既にインストールされているアプリを使用する場合は、**パッケージ/アクティビティ** リストからアプリを選択するか、**パッケージ/アクティビティ** フィールドにパッケージとアクティビティを指定します。
 - iOS デバイス上に既にインストールされているアプリを使用する場合は、**バンドル ID** を指定します。
 - UFT Mobile 上で利用可能なアプリを使用する場合は、**アプリ識別子** を指定します。
11. **OK** をクリックします。Silk Test Workbench が最小化され、アプリケーションと **記録中** ウィンドウが開きます。
12. テストする操作を記録します。
記録中は、タスクバーの Silk Test Workbench アイコンが点滅します。記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。
13. サポートするブラウザ上で Web アプリケーションを記録している場合で、かつ UI で直接アクセスできないコントロールを扱いたい場合 (たとえば、コントロールが他のコントロールで隠されている場合)、**記録中** ウィンドウの **階層ビューの切り替え** をクリックして、コントロール階層ツリーからコントロールを選択します。
この機能は、次のブラウザに対するテストで利用できます。
- Microsoft Edge
 - Apple Safari
 - Mozilla Firefox 41 以降
 - Google Chrome 50 以降
 - モバイル ブラウザー
14. 省略可能：検証ロジックをテストに追加します。
- 標準アプリケーションまたは Web アプリケーションをテストしている場合、テストに検証ロジックを追加するには、**Ctrl+Alt** をクリックします。Silk Test Workbench により記録が一時的に停止し、**テスト ロジック デザイナ** ウィザードが表示されます。

ウィザードの指示に従って処理を行い、**完了** をクリックしてウィザードを閉じ、記録を続行します。

- Microsoft Edge 上の Web アプリケーション、またはモバイル アプリケーションをテストしている場合、検証したいオブジェクトをクリックし、**操作の選択** ダイアログ ボックスで **検証の追加** をクリックします。

検証の追加に関する詳細については、「記録中のスクリプトへの検証の追加」を参照してください。

15 Alt+F10 を押すか、**記録中** ウィンドウで **停止** をクリックするか、Silk Test Workbench のタスクバー アイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、ビジュアルテストは **テスト ステップ** ペインに表示されます。

16 次のいずれか 1 つのステップを行います：

- **記録完了** ダイアログ ボックスを閉じて、テスト スクリプトを保存するには、**再生** をクリックします。
- **再生** ダイアログ ボックスが表示され、結果情報を指定してテストを再生できます。
- **ビジュアル ナビゲータ** の **テスト ステップ** ペインでビジュアルテストを開くには、**ビジュアルテストへ移動** をクリックします。
- ビジュアルテストを保存して、**記録完了** ダイアログ ボックスを終了するには、**保存** をクリックします。

ビジュアルテストが **ビジュアル ナビゲータ** の **テスト ステップ** ペインに表示されます。

記録されたテストが **ビジュアル ナビゲータ** に表示されているときには、いつでもそのテストを再生できます。

開始画面からのビジュアルテストの記録

記録中、セッションを停止するまで、Silk Test Workbench との対話を除くすべての操作が記録されます。記録が完了したら、ビジュアルテストを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. 省略可能 : Web アプリケーションに対してテストをする際に、画面プレビューから記録したコントロールを更新できるようにするには、ビジュアルテストを記録する前にブラウザ コントロールのキャプチャを有効化してください。

詳細については、「記録の出力オプションの設定」を参照してください。



注: Web アプリケーションの記録中にコントロールのキャプチャを有効化すると、ビジュアルテストのパフォーマンスが低下する恐れがあります。Micro Focus では、コントロールのキャプチャを無効化することをお勧めします。

2. **開始画面** で、**新しいビジュアルテストの記録** をクリックします。**アプリケーションの選択** ダイアログ ボックスが開きます。
3. 現在のプロジェクトに対してアプリケーション構成が設定されていない場合、テストするアプリケーションの種類に対応するタブを選択します。
 - ブラウザで実行しない標準アプリケーションをテストする場合は、**Windows** タブを選択します。
 - Web アプリケーションまたはモバイル Web アプリケーションをテストする場合は、**Web** タブを選択します。
 - ネイティブ モバイル アプリケーションをテストする場合は、**モバイル** タブを選択します。
4. 標準アプリケーションをテストするには、現在のプロジェクトに対してアプリケーション構成が設定されていない場合は、リストからアプリケーションを選択します。
5. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。

- a) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザのインスタンスが既に実行されている場合、**実行中のブラウザの URL を使用する** をクリックして、実行中のブラウザ インスタンスに現在表示されている URL の記録を行うことができます。
 - b) 省略可能：あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザ ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
6. ネイティブ モバイル アプリケーション (アプリ) をテストするには：
- a) アプリをテストするモバイル デバイスをリストから選択します。
 - b) ネイティブ モバイル アプリケーションを選択します。
 - モバイル デバイスまたはエミュレータ上にアプリをインストールする場合は、**参照** をクリックしてアプリ ファイルを選択するか、**アプリ ファイル** テキスト フィールドにアプリ ファイルへの完全パスを入力します。このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
 - Android デバイス上に既にインストールされているアプリを使用する場合は、**パッケージ/アクティビティ** リストからアプリを選択するか、**パッケージ/アクティビティ** フィールドにパッケージとアクティビティを指定します。
 - iOS デバイス上に既にインストールされているアプリを使用する場合は、**バンドル ID** を指定します。
 - UFT Mobile 上で利用可能なアプリを使用する場合は、**アプリ識別子** を指定します。
7. **OK** をクリックします。Silk Test Workbench が最小化され、アプリケーションと **記録中** ウィンドウが開きます。
8. テストする操作を記録します。
記録中は、タスクバーの Silk Test Workbench アイコンが点滅します。記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。
9. サポートするブラウザ上で Web アプリケーションを記録している場合で、かつ UI で直接アクセスできないコントロールを扱いたい場合 (たとえば、コントロールが他のコントロールで隠されている場合)、**記録中** ウィンドウの **階層ビューの切り替え** をクリックして、コントロール階層ツリーからコントロールを選択します。
この機能は、次のブラウザに対するテストで利用できます。
- Microsoft Edge
 - Apple Safari
 - Mozilla Firefox 41 以降
 - Google Chrome 50 以降
 - モバイル ブラウザー
- 10 Alt+F10 を押すか、**記録中** ウィンドウで **停止** をクリックするか、Silk Test Workbench のタスクバー アイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、ビジュアル テストは **テスト ステップ** ペインに表示されます。
- 11 次のいずれか 1 つのステップを行います：
- **記録完了** ダイアログ ボックスを閉じて、テスト スクリプトを保存するには、**再生** をクリックします。
再生 ダイアログ ボックスが表示され、結果情報を指定してテストを再生できます。
 - **ビジュアル ナビゲータ** の **テスト ステップ** ペインでビジュアル テストを開くには、**ビジュアル テストへ移動** をクリックします。

- ビジュアル テストを保存して、**記録完了** ダイアログ ボックスを終了するには、**保存** をクリックします。

ビジュアル テストが **ビジュアル ナビゲータ** の **テスト ステップ** ペインに表示されます。

記録されたテストが **ビジュアル ナビゲータ** に表示されているときには、いつでもそのテストを再生できます。

複数のテスト アプリケーションをテストするビジュアル テストの記録

複数のテスト アプリケーションをテストするビジュアル テストを記録できます。たとえば、データベースを変更するアプリケーションをテストしているときに、データベースの内容を確認するためにデータベースのビューアー ツールを使用する場合には、そのデータベースのビューアー ツール用のアプリケーション構成を追加する必要があります。

1. テストする主要なアプリケーション用に、ビジュアル テストを記録するか手動で作成します。
2. ビジュアル テストを開きます。
3. **タスク** ペインで **<<開始>>** ステップをクリックします。
4. **プロパティ** ペインで、**アプリケーション構成** カテゴリに移動します。
5. **アプリケーション構成** テキスト ボックスをクリックします。
6. **アプリケーション構成の追加** をクリックします。**アプリケーションの選択** ダイアログ ボックスが表示されます。
7. 追加でテストするアプリケーションの種類に対応するタブを選択します。
 - ブラウザで実行しない標準アプリケーションをテストする場合は、**Windows** タブを選択します。
 - Web アプリケーションまたはモバイル Web アプリケーションをテストする場合は、**Web** タブを選択します。
 - ネイティブ モバイル アプリケーションをテストする場合は、**モバイル** タブを選択します。
8. 標準アプリケーションをテストするには、リストからアプリケーションを選択します。
9. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。
 - a) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザのインスタンスが既に実行されている場合、**実行中のブラウザーの URL を使用する** をクリックして、実行中のブラウザ インスタンスに現在表示されている URL の記録を行うことができます。
 - b) 省略可能：あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。

たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザー サイズの編集** をクリックすると、新しいブラウザー サイズを指定したり、**ブラウザー サイズ** リストに表示するブラウザー サイズを選択することができます。
10. ネイティブ モバイル アプリケーション (アプリ) をテストするには：
 - a) アプリをテストするモバイル デバイスをリストから選択します。
 - b) ネイティブ モバイル アプリケーションを選択します。
 - モバイル デバイスまたはエミュレータ上にアプリをインストールする場合は、**参照** をクリックしてアプリ ファイルを選択するか、**アプリ ファイル** テキスト フィールドにアプリ ファイルへの完全パスを入力します。このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
 - Android デバイス上に既にインストールされているアプリを使用する場合は、**パッケージ/アクティビティ** リストからアプリを選択するか、**パッケージ/アクティビティ** フィールドにパッケージとアクティビティを指定します。

- iOS デバイス上に既にインストールされているアプリを使用する場合は、**バンドル ID** を指定します。
- UFT Mobile 上で利用可能なアプリを使用する場合は、**アプリ識別子** を指定します。

11OK をクリックします。Silk Test Workbench により、新しいアプリケーション構成が **プロパティ** ペインに追加されます。

12新しいアプリケーション構成を使用して、ビジュアル テストの追加ステップを記録します。

ビジュアル テストでのロケーターからオブジェクト マップ エントリへの移動

オブジェクト マップ エントリの **ID** 以外の情報、つまりコマンドが実行されたときに Open Agent が使用する生のロケーターを参照したい場合は、以下の手順に従います。

1. ビジュアル テストを開きます。
2. **テストステップ** ペインでステップを選択します。 **プロパティ** ペインで、**ロケーター** フィールドにロケーター テキストが表示されます。
3. カーソルを **ロケーター** フィールドのテキストに置いてください。 **オブジェクト マップを開く** ボタンが右側に表示されます。
4. **オブジェクト マップを開く** ボタンをクリックします。 選択された適切な項目が表示された状態で、**オブジェクト マップ** ウィンドウが、ツリー ビューに表示されます。

Java Network Launching Protocol (JNLP) を使用するアプリケーションを起動するビジュアル テストの構成

Java Network Launching Protocol (JNLP) を使用して起動するアプリケーションでは、Silk Test Workbench に追加の構成が必要です。これらのアプリケーションは Web から起動されるため、実際のアプリケーションと、「Web Start」を起動するアプリケーションを起動するように、アプリケーション構成を手動で構成する必要があります。このようにしない場合、アプリケーションがすでに実行されていないがぎり、ビジュアル テストを再生すると、失敗します。

1. テストするアプリケーションのビジュアル テストを記録します。
2. **タスク** ペインで <<開始>> ステップをクリックします。
3. **プロパティ** ペインで、**アプリケーション構成** カテゴリに移動します。
4. **アプリケーション構成** フィールドをクリックし、**アプリケーション構成の追加** をクリックします。 **アプリケーションの選択** ダイアログ ボックスが表示されます。
5. **Windows** タブを選択します。
6. テストする Java アプリケーションを選択します。
7. **OK** をクリックします。Silk Test Workbench は新しいアプリケーション構成を追加します。
8. javaws.exe を使用して Web Start をダウンロードし、起動するように、最初のアプリケーション構成 (**構成 1**) を変更します。
 - a) **構成 1** フィールドをクリックし、**アプリケーション構成の編集** をクリックします。
 - b) **実行可能ファイル** フィールドに、Web Start の実行可能ファイルの名前とファイルへのパスを入力します。
たとえば、以下のように入力します。 %ProgramFiles%\Java\jre6\bin\javaws.exe
 - c) **作業ディレクトリ** フィールドに、ダウンロードするファイルの一時ディレクトリの場所を入力します。
たとえば、C:\temp と入力します。 temp は、ローカル C ドライブのディレクトリです。
 - d) **ロケーター** フィールドには、テストするアプリケーションのメイン ウィンドウを識別するためのロケーターを指定します。
ビジュアル テストを記録したときに正しいロケーターが記録されているため、ほとんどの場合、これを変更する必要はありません。

たとえば、SwingSet3 JNLP アプリケーションをテストする場合、ロケータは SwingSet3 です。

e) **コマンドライン引数** フィールドに、テスト アプリケーションを起動する JNLP の URL を入力します。

たとえば、SwingSet3 JNLP アプリケーションを使用するには、`http://download.java.net/javadesktop/swingset3/SwingSet3.jnlp` と入力します。

f) **コマンドラインパターン** フィールドのすべての値を削除します。

g) 基本状態に指定されたロケータの Silk Test Workbench による検索が何ミリ秒経過後にタイムアウトするかを示す値を変更するには、**タイムアウト** フィールドに値を入力します。

デフォルトで、**タイムアウト (ミリ秒)** は 30000 に設定されています。ほとんどのアプリケーションでは、この値で十分です。

h) **基本状態の実行** テキスト ボックスは **True** に設定されたままにします。

9. テストする Java アプリケーションを使用するように、新しいアプリケーション構成を変更します。

a) **実行可能ファイル** フィールドに、テストするアプリケーションの実行可能ファイルの名前とファイルへのパスを入力します。

たとえば、SwingSet3 JNLP アプリケーションを使用するには、`*%javaw.exe` と入力します。

b) **作業ディレクトリ** フィールドは空のままにします。

c) **ロケータ** フィールドには、テストするアプリケーションのメイン ウィンドウを識別するためのロケータを指定します。

これは、ほとんどの場合、**構成 1** で指定したロケータ名と一致します。

たとえば、SwingSet3 JNLP アプリケーションをテストする場合、ロケータは SwingSet3 です。

d) **コマンドライン引数** フィールドのすべての値を削除します。

e) **コマンドラインパターン** フィールドに、テスト アプリケーションのコマンドライン引数のパターンを入力します。

コマンドライン引数のパターンを含むアプリケーション構成では、プロセス名とコマンドライン引数のパターンの両方が一致するプロセスのみが、テストに対して有効化されます。コマンドライン引数のパターンが定義されていない場合、指定されたプロセス名の実行可能パターンを持つプロセスがすべて有効化されます。

たとえば、SwingSet3 JNLP アプリケーションを使用するには、`*SwingSet*` と入力します。この引数により、SwingSet3 Java アプリケーションのみが確実に起動されます。

f) 基本状態に指定されたロケータの Silk Test Workbench による検索が何ミリ秒経過後にタイムアウトするかを示す値を変更するには、**タイムアウト** テキスト ボックスに値を入力します。

デフォルトで、**タイムアウト (ミリ秒)** は 30000 に設定されています。ほとんどのアプリケーションでは、この値で十分です。

g) **基本状態の実行** チェック ボックスをオンにします。

ビジュアルテストを再生すると、JNLP アプリケーションが予期したとおりに起動します。

新しいブラウザ ウィンドウで開くビジュアルテストの構成

ビジュアルテスト用に既存のブラウザ ウィンドウを使用するのではなく、別のブラウザ ウィンドウを開く場合があります。たとえば、Silk Central Test Manager から Silk Test Workbench ビジュアルテストを実行すると、テストは既存の Silk Central ウィンドウを使用します。Silk Central に戻るには、ユーザーはブラウザ ウィンドウで **戻る** をクリックする必要があります。Silk Central から実行するテスト用に別のブラウザ ウィンドウを開くテストを構成すると、この動作を回避できます。

1. テストするアプリケーションのビジュアルテストを記録します。
2. **テストステップ** ペインで、**<<開始>>** ステップをクリックします。
3. **プロパティ** ペインで、**アプリケーション構成** カテゴリに移動します。
4. **アプリケーション構成** フィールドをクリックし、**アプリケーション構成の追加** をクリックします。**アプリケーションの選択** ダイアログ ボックスが表示されます。

5. アプリケーション構成をクリックし、**アプリケーション構成の編集** をクリックします。**アプリケーション構成の編集** ダイアログ ボックスが表示されます。
6. **ブラウザーの種類** リストで、使用するブラウザーを選択します。
7. **基本状態の実行** チェック ボックスをオフにします。
8. **テスト ステップ** ペインで、**挿入 > 外部プログラム** を選択します。Silk Test Workbench によって、「プログラム」の実行 (通常表示) が **テスト ステップ** ペインに追加されます。
9. **プロパティ** ペインで、**コマンド** カテゴリに移動します。
10. **パス名** フィールドをクリックし、**ファイルの選択** をクリックします。**ファイルの選択** ダイアログ ボックスが表示されます。
11. **ブラウザーの実行可能ファイル**へ移動し、**開く** をクリックします。
たとえば、Internet Explorer を使用するには、C:\Program Files\Internet Explorer\iexplore.exe を指定します。
12. **操作** リスト ボックスで **開く** が指定されていることを確認します。
13. テストする Web アプリケーションの URL を、**パラメータ** フィールドに入力します。
たとえば、サンプル Web アプリケーションをテストするには、http://demo.borland.com/InsuranceWebExtJS/index.jsf を入力します。

ビジュアル テスト内での特定のステップの表示

ビジュアル テストで表示するテスト ステップをカスタマイズして、特定のステップだけを表示することができます。ビジュアル テストに表示されるステップをフィルタ処理すると、テストの性質をわかりやすくしたり、レビューが必要なステップのみにフォーカスしたりできます。特定のタイプのステップのみが表示されているビジュアル テストで追加の操作を記録すると、記録後は表示条件を満たす記録済み操作のみが表示されます。表示オプションを **ステップと画面** に変更すると、新しく記録した他のステップが表示されます。


1. ビジュアル テストで **テスト ステップ** ペインを開き、**操作 > 表示** を選択します。
2. ビジュアル テストで以下のステップ表示オプションを選択します。
3. 画面のみを操作するビジュアル テストのステップをすべて非表示にする場合は、**ステップのみ** をクリックします。
4. テスト操作を実行するステップをすべて非表示にする場合は、**画面のみ** をクリックします。
5. ビジュアル テストのテスト ステップをすべて表示する場合は、**ステップと画面** をクリックします。
これはデフォルトの表示設定です。
6. **テスト ステップ** ペインに **説明** 列を表示するには、**ステップの説明** をクリックします。この列は、デフォルトでは表示されません。説明が記述されたステップには、**説明** 列に **テスト ステップの説明** のアイコンが表示されます。**プロパティ** ペインでステップの **説明** の値を更新すると、テスト ステップに説明を入力できます。

テスト アプリケーションでのビジュアル テスト用コントロールの識別


テストを記録する場合、Silk Test Workbench によって自動的にコントロールが識別されます。ビジュアル テストを記録した後で、テスト対象アプリケーションで可視のコントロールを識別して、ビジュアル テストで使用するコントロールを変更したり、ビジュアル テストに手動でテスト ステップを挿入することができます。

1. **テスト ステップ** ペインで、コントロールを変更するステップを選択します。
2. **プロパティ** ペインで、**ローケータ** テキスト ボックスの **コントロールの識別** メニューを展開します。
3. **テスト対象アプリケーション** を選択します。
4. テストしているアプリケーションにカーソルを移動します。

カーソルをアプリケーション内の各コントロール上に動かすたびにコントロールが濃い枠で強調表示されます。

 **ヒント:** Esc を押すと、識別モードが終了し、ビジュアルテストに戻ります。


5. 使用するコントロールが強調表示されているときに、コントロールをクリックします。

 **注:** 別のクラスにアクセスしている自動ステップを編集した場合、ステップによってアクセスされているクラスが変更されたこと、またその変更を適用するかどうかを確認するメッセージが Silk Test Workbench から表示されます。**はい** をクリックして新しいクラスとコントロール情報で **ロケーター** を更新するか、**いいえ** をクリックしてコントロールを識別せずに **ビジュアルナビゲータ** に戻ります。


ロケーター テキストボックスが新たに選択されたコントロール情報に更新され、このコントロールが再生時に使用されます。

画面プレビューでのビジュアルテスト用コントロールの識別

テストを記録する場合、Silk Test Workbench によって自動的にコントロールが識別されます。ビジュアルテストを記録し、テスト対象アプリケーションが利用できない場合、**画面プレビュー** を使用してビジュアルテストで使用するコントロールを変更したり、ビジュアルテストに手動でテストステップを挿入することができます。


 **注:** Web アプリケーションに対して、ビジュアルテストの**画面プレビュー** でコントロール名を識別するには、ビジュアルテストの記録時にブラウザ コントロールのキャプチャが有効化されている必要があります。詳細については、「記録の出力オプションの設定」を参照してください。

1. **テストステップ** ペインで、コントロールを変更するステップを選択します。
2. **プロパティ** ペインで、**ロケーター** テキストボックスの **コントロールの識別** メニューを展開します。
3. **画面プレビュー** を選択します。
4. **画面プレビュー** でキャプチャされた画面にカーソルを移動します。

 **ヒント:** **画面プレビュー** で Esc を押すと、識別モードが終了し、ビジュアルテストに戻ります。

カーソルを **画面プレビュー** 内の各コントロールに動かすたびにコントロールが濃い枠で強調表示されます。

5. 使用するコントロールが強調表示されているときに、コントロールをクリックします。

 **注:** 別のクラスにアクセスしている自動ステップを編集した場合、ステップによってアクセスされているクラスが変更されたこと、またその変更を適用するかどうかを確認するメッセージが Silk Test Workbench から表示されます。**はい** をクリックして新しいクラスとコントロール情報で **ロケーター** を更新するか、**いいえ** をクリックしてコントロールを識別せずに **ビジュアルナビゲータ** に戻ります。

ロケーター テキストボックスが新たに選択されたコントロール情報に更新され、このコントロールが再生時に使用されます。

オブジェクトの識別ダイアログを使用したビジュアルテスト用コントロールの識別

ビジュアルテストを記録する場合、Silk Test Workbench によって自動的にコントロールが識別されます。ビジュアルテストを記録した後で、ビジュアルテストで使用するコントロールを変更したり、ビジュアルテストに手動で追加のテストステップを挿入することができます。**オブジェクトの識別** ダイアログを使用して、コンテナのようなテスト対象アプリケーションの非表示コントロールを識別できます。

1. **テストステップ** ペインで、コントロールを変更するステップを選択します。
2. **プロパティ** ペインで、**ロケーター** テキスト ボックスの **コントロールの識別** メニューを展開します。
3. **オブジェクトの識別ダイアログ** を選択します。
4. Web アプリケーションをテストする場合は、**ブラウザーの選択** ダイアログ ボックスが開きます。コントロールを識別するブラウザーを選択し、**識別の開始** をクリックします。**オブジェクトの識別** ダイアログ ボックスが開きます。
5. **識別の開始** をクリックします。
6. テストしているアプリケーションにカーソルを移動します。カーソルをアプリケーション内の各コントロール上に動かすたびにコントロールが強調表示されます。関連するロケーター文字列またはオブジェクト マップ項目は、**選択済みロケーター** テキスト ボックスに表示されます。
7. 使用するコントロールが強調表示されているときに、コントロールをクリックします。
8. 省略可能：**ロケーターの詳細の表示** をクリックすると、ロケーター ツリーや関係する属性のすべてが **ロケーター属性** テーブルに表示されます。
9. 省略可能：記録したロケーター属性は、**ロケーター属性** テーブルの別のロケーター属性で置き換えることができます。
たとえば、記録したロケーターは以下のように表示されます。
`/BrowserApplication//BrowserWindow//input[@id='loginButton']`
ロケーター属性 テーブルに textContents Login がリストされている場合、以下のようにしてロケーターを手動で変更できます。
`/BrowserApplication//BrowserWindow//input[@textContents='Login']`
新しいロケーターは、**選択済みロケーター** テキスト ボックスに表示されます。
- 10 **テスト** をクリックして、ロケーターがコントロールを正しく認識することを確認します。
- 11 ロケーターを使用する新しいステップをビジュアル テストに追加するには、**貼り付け** をクリックします。**操作の選択** ダイアログ ボックスが開きます。コントロールに対して実行する操作を選択し、**OK** をクリックします。

特定のテスト ステップでの追加操作の記録

ビジュアル テスト ステップを作成したあと、テストを開き、テストの任意の場所から追加操作を記録できます。これにより、いつでも既存のテストを追加操作で更新できます。

1. 既存のビジュアル テストを開きます。
2. 追加操作を記録する場所の直後のテスト ステップを選択します。



注: 記録した操作は、選択した場所の後に挿入されます。テスト対象アプリケーション (AUT) は基本状態に戻りません。代わりに、テスト スクリプトの直前の操作が記録された範囲で AUT を開いておきます。

3. **操作 > 記録** をクリックします。

Silk Test Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。

4. AUT に対して実行したい追加操作を記録します。

記録中は、タスクバーの Silk Test Workbench アイコンが点滅します。記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。

5. 記録を停止するには、**記録中** ウィンドウの **停止** をクリックします。

記録を停止するキーの組み合わせ (デフォルトでは **Alt+F10**) を押すか、タスクバーの Silk Test Workbench アイコンをクリックして、記録を停止することもできます。

テスト アプリケーションからビジュアル テストへの画面の挿入

ビジュアル テスト ステップを作成したあと、追加の画面と操作を記録するのではなく、テストを開き、画面キャプチャを挿入して、関連するテスト ステップを手動で追加することができます。画面キャプチャを挿入すると、テスト アプリケーションの更新された画面を含むビジュアル テストを手動で構築できます。この方法を使用すると、新しい画面を記録するよりも効率が上がる場合があります。

1. テストするアプリケーションを開き、キャプチャするページまたはダイアログに移動します。
2. 既存のビジュアル テストを開きます。
3. 新しい画面の挿入箇所の直後にあるテスト ステップを選択します。



注: テスト ステップと画面を **テスト ステップ** ペインに表示するには、**操作 > 表示 > ステップと画面** をクリックします。

4. **挿入 > Using** を選択します。

Silk Test Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。

5. キャプチャする画面をクリックするか、またはホット キーのキー入力の組み合わせを押して、画面をキャプチャします。

クリックまたはホット キーのキー入力の組み合わせによって画面がキャプチャされるかどうかは、ロケータの記録設定オプションによって決まります。この設定を変更するには、**ツール > オプション** を選択し、**オプション** メニュー ツリーの **記録** を展開して、**ロケーター** をクリックします。

新しい Using ステップが、**テスト ステップ** ペインで、選択したテスト ステップのあとに表示されます。

必要に応じて、関連付けられている自動テスト ステップを追加します。

ビジュアル テストのオブジェクト マップ項目またはロケーターの手動による記録

開始する前に、テストするアプリケーションが実行中であることを確認します。

オブジェクトの識別 ダイアログ ボックスを使用して、オブジェクト マップ項目またはロケーターを手動でキャプチャし、ビジュアル テストに対応するテスト ステップを作成します。

1. 変更するビジュアル テストを開きます。記録された各操作に対応するステップは、**テスト ステップ** ペインに表示されます。
2. オブジェクトと操作を挿入するテスト ステップをクリックします。
新しいオブジェクトと挿入は、選択したテスト ステップの下に挿入されます。
3. **ツール > オブジェクトの識別** を選択します。**オブジェクトの識別** ダイアログ ボックスが開きます。
4. **選択モード** を指定します。

- **クリック** : オブジェクトをクリックして、ロケーターを識別します。
- **ホット キー** : **キー入力** リスト ボックスで指定されたキー入力の組み合わせを使用してオブジェクトをキャプチャするには、このモードを指定します。通常、このモードは、ユーザーがクリックしたときにだけ表示されるメニューやコンボ ボックスなどのオブジェクトをキャプチャする場合に選択します。このモードでは、オブジェクトを選択し、ホット キー入力の組み合わせを押すことによって、オブジェクトを閉じることなくロケーターをキャプチャします。

5. **識別の開始** をクリックします。

Google Chrome でオブジェクト マップ項目またはロケーターをキャプチャする場合は、Google Chrome の選択されたインスタンスが適切なオートメーション パラメータを使用して起動されているかどうかを Silk Test Workbench が検出します。適切なパラメータを使用して起動されていない場合

は、Silk Test Workbench が Google Chrome を終了し、オートメーションパラメータセットを使用して再起動します。

6. 省略可能：ロケータを記録する前に、テスト対象アプリケーションを特定の状態にするには、**識別の停止** をクリックします。これ以降、テスト対象アプリケーションで実行する操作は、記録されなくなります。ロケータの記録を再開する場合は、**識別の開始** をクリックします。
7. 記録したいオブジェクト上にマウスを移動して、次のステップのいずれかを行います：
 - **クリック** モードを使用する場合は、識別するオブジェクトをクリックします。
 - キー入力の組み合わせを押して、**ホットキー** モードでオブジェクトをキャプチャします。
デフォルトでは、キー入力の組み合わせは Ctrl+Shift です。

Silk Test Workbench によって、関連するロケータ文字列が **選択済みロケータ** テキストボックスに表示されます。

8. **ロケータの詳細** テーブルでのロケータの表示方法を絞り込むには、以下のいずれかの操作を実行します。
 - **ロケータの詳細の非表示**：**ロケータの詳細** テーブルを非表示にするには、このリンクをクリックします。
 - **オブジェクト マップの表示**：オブジェクト マップ項目名を **ロケータ** 列に表示するには、このチェックボックスをオンにします。オブジェクト マップ項目名は、コントロールまたはウィンドウに対して、コントロールやウィンドウのロケータではなく論理名 (エイリアス) を関連付けます。デフォルトでは、オブジェクト マップ項目名が表示されます。ロケータを使用するには、このチェックボックスをオフにします。
 - **完全ロケータの表示**：完全ロケータ名を表示するには、このチェックボックスをオンにします。オブジェクトに関連付けられた属性のみを表示するには、このチェックボックスをオフにします。
 - **プロパティの表示**：**ロケータの詳細** テーブルで選択したオブジェクトの属性および属性値を表示するには、このチェックボックスをオンにします。このテーブルの属性を選択して、ロケータの識別に使用できます。[プロパティ] サブツリーを非表示にし、ロケータの詳細のみを表示するには、このチェックボックスをオフにします。
9. **選択済みロケータ** テキストボックスに表示されるオブジェクトが使用するオブジェクトであることをテストするには、**テスト** をクリックします。Silk Test Workbench により、ロケータに対応するオブジェクトが、テストするアプリケーションで強調表示されます。
- 10 記録したロケータを置換するには、**ロケータの詳細** テーブルで、使用するロケータを選択します。新しいロケータは、**選択済みロケータ** テキストボックスに表示されます。
- 11 オブジェクトをコピーし、ビジュアルステップでテストステップを作成するには、**貼り付け** をクリックします。**操作の選択** ダイアログボックスが開きます。
- 12 オブジェクトで実行する操作を選択し、**OK** をクリックします。Silk Test Workbench により、新しいテストステップのオブジェクトと操作が、ビジュアルテストに追加されます。

既存のビジュアルテストを開く


既存のビジュアルテストは、**開始画面** または **アセットブラウザ** から開くことができます。

1. **開始画面** からビジュアルテストを開くには、**開始画面** の **スクリプト** ペインで以下のステップのいずれかを実行します。
 - **最新** リストで、適切なビジュアルテストの **ビジュアルテスト 'プロジェクト名 - テスト名'** を開くをクリックします。
 - **開く** をクリックして **アセットブラウザ** を開き、適切なビジュアルテストを右クリックして **開く** を選択します。
2. **アセットブラウザ** からビジュアルテストを開くには、以下のステップを実行します。
 - a) **表示** > **アセットブラウザ** を選択します。
 - b) **資産の種類** ペインで **ビジュアルテスト** を選択し、適切なビジュアルテストを右クリックして **開く** を選択します。

ビジュアルナビゲータのテストステップペインで、ビジュアルテストが開きます。

ビジュアルテストの保存

新しいビジュアルテストまたは変更したビジュアルテストは明示的に保存することができます。更新したビジュアルテストを終了するときに、変更を保存するかどうかを確認するメッセージが表示されます。

 **ヒント:** 保存した変更を取り消すことはできません。

1. **ファイル > 名前を付けて保存** を選択します。 **名前を付けて保存** ダイアログボックスが開きます。
2. 省略可能：ビジュアルテスト名を変更し、説明を入力します。
3. **OK** をクリックします。

ビジュアルテストの概要

ビジュアルテストは、自動テストソリューションの基本要素です。Silk Test Workbench では、アプリケーションのテスト中に実行する操作をビジュアルテストを使用して模倣します。ユーザーがキー入力やマウス操作でメニュー、リスト項目、ボタンなどを選択するのと同じように、アプリケーションや Web ページを制御します。ビジュアルテストを記録すると、Silk Test Workbench によりキー入力とマウスクリックがすべて生成されます。

メニューの選択、データの入力、データ処理の確認など、アプリケーションのテストに使用する操作は、ビジュアルテストの個々のステップで表されます。各ステップがビジュアルテスト内の 1 行になります。記録したステップは、変更したり、再生したりできます。

Silk Test Workbench では、ビジュアルテストをすばやく記録して再生することができます。ビジュアルテストを変更して、記録できないステートメントを追加したり、テストアプリケーションの変更を反映した修正を加えたりできます。また、既存のビジュアルテストからステップをカットアンドペーストして新しいビジュアルテストを作成することもできます。

ビジュアルテストを自動的に記録したり、手動で編集したり、または、自動と手動のテストを組み合わせで使用したりできます。

Silk Test Workbench VB .NET スクリプトには、一連のコード行が含まれ、ビジュアルテストには一連のテストステップが含まれます。コードの各行または各テストステップは、テストアプリケーションに対する個々の操作を実行します。

ビジュアルテストの利点

ビジュアルテストには以下のような利点があります。

- プログラミング言語に基づくスクリプトを生成せずに、テストアプリケーションに対してテストを記録し、編集できる。
- テストステップレベルで、視覚的にテストアプリケーションのコントロールに対して操作と値を設定できる。
- ビジュアルテストのテキストベースの操作ステップとその操作のオブジェクトの間の関係を参照できる。
- 自動処理されるビジネスプロセスに精通しているユーザーにとって、テストプロセスでのオーナーシップが大きくなる。
- 既存のスクリプトや他のすべての既存のテスト資産をビジュアルテストで使用するにより、有効活用できる。
- テスト対象のアプリケーションにアクセスしなくても、テストアプリケーションとコントロールを再キャプチャできる。
- テストを作成し直さなくても、テストアプリケーションの更新に従って画面キャプチャを自動的に更新できる。
- フラグを使用して、テストプロジェクトの問題、割り当て、その他のテストチーム情報に関して、協業/共有できる。

- ビジュアル テストを再生することでアプリケーションをテストできる。ビジュアル テストは、アプリケーション テストのためにいつでも再生できる。ビジュアル テストでは、以下のことが可能。
 - 指定した時刻に自動的に実行する。
 - 完全に無人で実行する。
 - ユーザーとの対話を行って、ID やパスワードなどの認証情報を受け取ることができる。

スクリプトと比較したビジュアル テストの利点

スクリプトと同様に、ビジュアル テストはアプリケーションのテストに使用する操作を記録します。これらの操作には、メニューの選択、データの入力、アイコンのクリック、結果の確認などが含まれます。ビジュアル テストは Silk Test Workbench の主要なテスト単位です。テスト ソリューションにスクリプトだけを使用する場合と比較して、以下のような利点があります。

- ビジュアル テストでは、一連のステップが技術的ではない明確な言語で表示されるため、テスト担当者が容易に理解できます。
- ビジュアル テストはスクリプト言語に縛られないため、プログラミング言語の知識がないテスト担当者でも、詳細で強力なテスト ソリューションを作成できます。
- ビジュアル テストはストーリーボードにサムネイル イメージでグラフィカルに表示されるため、ビジュアル テストの全体のフローをすぐに確認できます。
- ビジュアル テストを記録したあとは、**画面プレビュー** を使用して、テスト アプリケーションにアクセスしなくても、そのビジュアル テストを更新できます。
- ビジュアル テストはテスト ソリューションのビルド プロセスを円滑にします。たとえば、1 人のテスト担当者がテストを記録してそれを別のテスト担当者に渡します。そのテスト担当者はロジック、検証、ループの他、予期しない状況に特定の方法で対処する機能を加えて、ビジュアル テストに情報を追加します。
- ビジュアル テストは、テスト アプリケーションが変わった場合に、作成し直す必要がありません。テスト ステップをすぐに更新できます。
- 既存のビジュアル テストをコピーして変更することにより、新しい手順を簡単に作成できます。
- テストへのモジュール式アプローチが推進され、再利用率を高めることができます。他のビジュアル テストを呼び出せます。
- Silk Test Workbench では、他のテスト担当者とのテスト プロジェクト情報のやり取りや協業ができます。

サンプル ビジュアル テスト

以下のサンプル ビジュアル テストでは、ビジュアル テストの一般的な機能を使用する方法を示しています。

サンプル ビジュアル テスト：アクティブ データの使用

このサンプル ビジュアル テストでは、ビジュアル テストにおけるアクティブ データの使用を示します。このビジュアル テストでは、既存のアクティブ データを使用し、**ListView** コントロールに複数の名前（姓と名）を入力することによって Windows テスト アプリケーションをテストします。ファイル内の各レコードには、姓と名が含まれています。サンプル ビジュアル テストはファイル内のデータを循環し、アプリケーションの **ListView** コントロールにデータを入力することによって Windows アプリケーションをテストします。

サンプル ビジュアル テスト：アクティブ データの使用：パート 1 シナリオ

テスト シナリオ

テスト担当者は、スプレッドシートに保存された顧客の姓と名のデータを使用して、Windows テスト アプリケーションに複数の名前レコードを入力します。テスト時に、データは Names1.xls という名前のスプレッドシートにあります。

これを行うには、以下を実行します。

- Windows テスト アプリケーションの **ListView** コントロールへのアクセスを記録し、コントロールで [First Name] および [Last Name] 列を作成してから、最初の姓と名のペアを入力します。
- ビジュアル テストに関連付けるアクティブ データ資産を作成します。アクティブ データ資産の作成時に、使用するデータが含まれる Names1.xls データ ファイルを選択します。
- そのアクティブ データ ファイルをビジュアル テストに関連付けます。
- 繰り返しロジックを作成して、テスト アプリケーションの **ListView** コントロールに姓と名を入力する記録済みステップを繰り返します。
- 姓と名を入力するステップのプロパティを変更して、アクティブ データ ファイルのデータが使用されるようにします。

サンプル ビジュアル テスト : アクティブ データの使用 : パート 2 詳細

サンプル ビジュアル テスト

以下の図に、このテスト シナリオに基づいた **テスト ステップ** ペインのサンプル ビジュアル テスト ステップを示します。テスト内のステップの進行を強調表示しています。

Test Steps		Actions
1	<<Start>>	
2	Using Window 'Test Application'	
3	Select menu item 'List view'	
4	Using Dialog 'List View'	
5	Enter 'First Name'	
6	Using Dialog 'List View'	
7	Select button 'Add Column'	
8	Using Dialog 'List View'	
9	Set selection range	
10	Using Dialog 'List View'	
11	Enter 'Last Name'	
12	Using Dialog 'List View'	
13	Select button 'Add Column'	
14	Repeat using activedata 'NamesList'	
15	Using Dialog 'List View'	
16	Enter '[[NamesList].Text("First Name")]'	
17	Using Dialog 'List View'	
18	Select button 'Add Item'	
19	Using Dialog 'List View'	
20	Enter '[[NamesList].Text("Last Name")]'	
21	Using Dialog 'List View'	
22	Select button 'Add Item'	
23	End Repeat	
24	<<End>>	

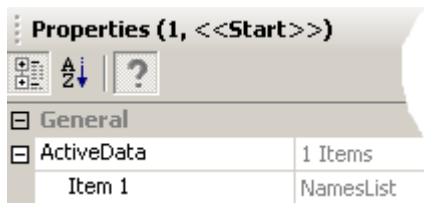
ステップの詳細

ビジュアルテストの各ステップの詳細は、以下のとおりです。

ステップ 1: <<開始>> ステップ。このステップのプロパティには、ビジュアル テストで使用されるアクティブ データ資産の名前が表示されます。

このテストシナリオでは、アクティブ データ カテゴリの **ステップ 1 項目 1** プロパティにアクティブ データ資産の関連付けが表示されます。このアクティブ データ資産には、ファイル Names1.xls が含まれています。ビジュアルテストでは、このファイルのデータが使用されます。

以下の図に、このビジュアルテストの <<開始>> ステップの **項目 1** プロパティを示します。



ステップ 2 から 3: テスト アプリケーションで **List view** メニュー項目を選択して **ListView** にアクセスする、記録されたステップ。このステップのあと、繰り返しステップが開始されます。

ステップ 4 から 13: **ListView** で First Name と Last Name の各列を作成する、記録されたステップ。これらの列は、アクティブ データ ファイルから適切な列に姓と名を入力するためのテストを設定します。

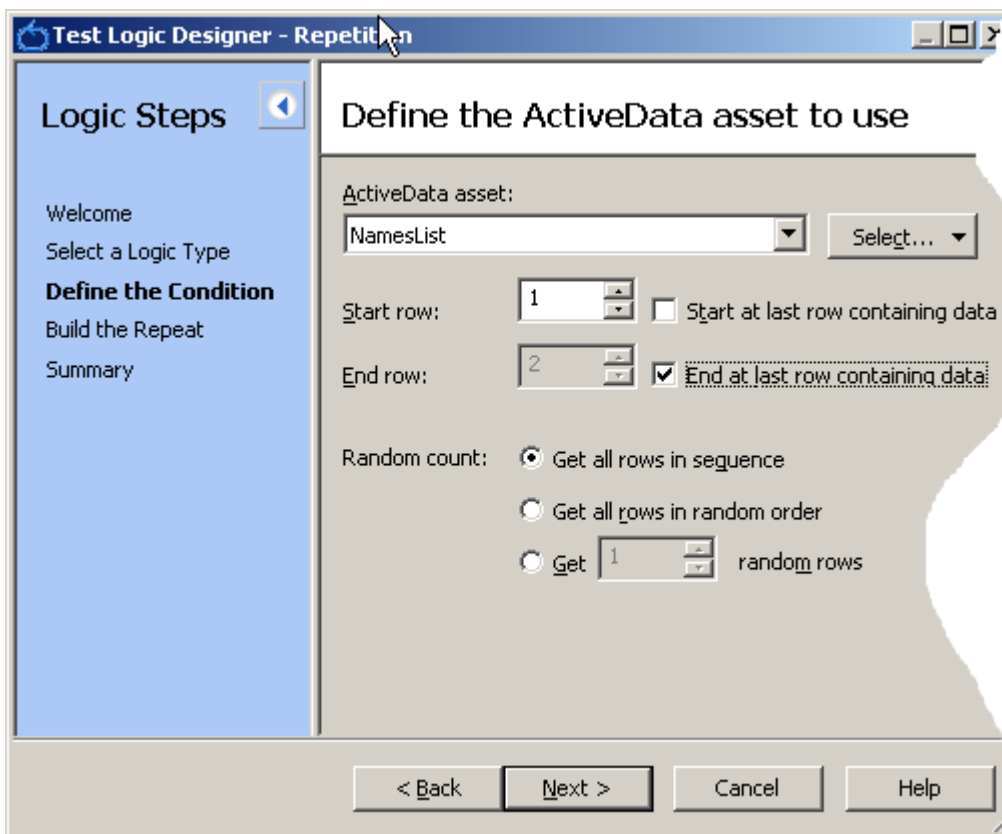
- 説明が「Using Dialog」で始まるステップでは、データを入力するためのウィンドウおよびコントロールにテストが関連付けられます。
- このシーケンスの他のステップは、列に名前を付け、それを **List view** に追加します。

ステップ 14: 「アクティブ データを使用して繰り返します」ステップでは、繰り返しロジックが開始します。テスト担当者は、テストを記録したあとで、このステップを作成します。このステップと繰り返しの終了ステップ（ステップ 23）の間にあるすべてのステップは、アクティブ データ ファイルの名前レコードごとに繰り返されます。使用する繰り返しロジックの種類は、**アクティブ データ ファイルを使用してステップ シーケンスを繰り返す** です。

繰り返し回数は、繰り返しロジックを作成したときに入力した **開始行**、**終了行**、および **ランダム数** の値によって決まります。また、この設定によって、アクティブ データ テストに使用するアクティブ データ ファイルのレコードも決まります。

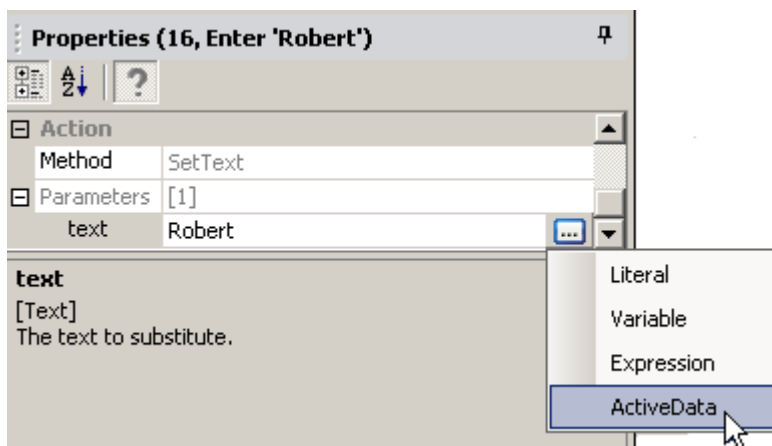
このテストシナリオでは、データ ファイル内のすべての行を順番に使用して、アクティブ データ テストを行います。データ ファイルに 10 行のデータが含まれており、すべてが使用される場合、繰り返しロジックステップの繰り返し回数は 10 回になります。

以下の図に、このビジュアルテストの繰り返しロジックを作成したときの **テスト ロジック デザイナ** の使用する **アクティブ データ資産の定義** ページを示します。

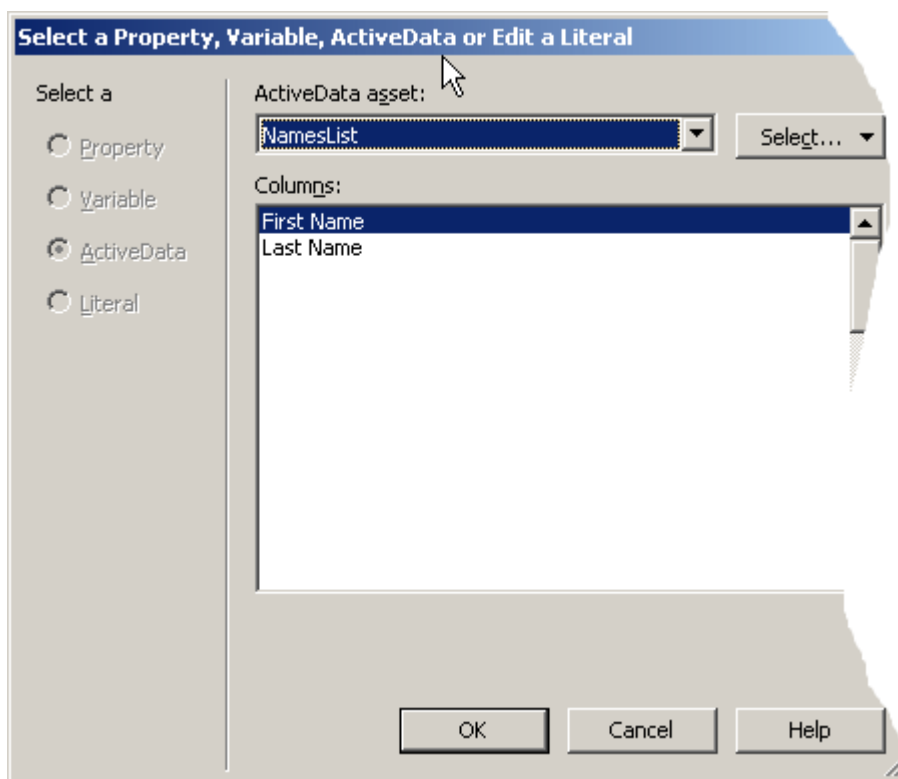


ステップ 15 から 22 : アクティブ データ ファイルのデータを使用するように記録後に変更されたステップ。テスト アプリケーションの **ListView** コントロールで、[First Name] と [Last Name] の列にデータを入力します。

アクティブ データ ファイルのデータを使用するようにこれらのステップを変更すると、リテラル値ではなくアクティブ データ値が使用されるように入力値が更新されます。これを行うには、以下の図に示すように、ステップの **Text** プロパティの値領域をクリックして、メニューから **アクティブ データ** を選択します。



ダイアログ ボックスが表示され、データ ファイルを含むアクティブ データ資産を選択します。また、**Text** プロパティの入力値にマップする値を含むデータ ファイル内の列を選択します。このシナリオのステップ 16 では、Windows テスト アプリケーションの [First Name] のリテラル テキストが、アクティブ データ ファイルの **名 (First Name)** 列のデータで置き換えられます。以下の図に、このダイアログ ボックスにおける設定を示します。



この変更により、ステップ テキストには以下が表示されるようになります。

- テストで使用されるアクティブ データ資産の名前
- 使用されるデータ型 (テキスト)
- ステップで使用されるデータが含まれる列の名前

アクティブ データ ファイルのデータを使用した自動テスト ステップの変更の詳細については、「アクティブ データ ファイルのデータのビジュアル テスト データへのマッピング」を参照してください。

ステップ 23 : ビジュアル テストの繰り返しロジックの一部として作成された 繰り返しの終了 ステップ。このステップで、繰り返しロジックのループが閉じます。

ステップ 24 : <<終了>> ステップ。ビジュアル テストの再生を完了します。

サンプル ビジュアル テスト : コントロールの選択を確認するための判断ロジックの使用

このサンプル ビジュアル テストでは、Windows サンプル アプリケーションで **ComboBox** コントロールが選択されていることを確認します。テスト対象の **ComboBox** がレンダリングされたあとに選択が行われていない場合、**ComboBox** にはテキストが表示されません。テスト担当者は、判断ロジックを使用して、**ComboBox** のテキストが null であるかどうかを判断し、その結果に応じて異なる操作を実行します。

サンプル ビジュアル テスト : コントロールの選択を確認するための判断ロジックの使用 : パート 1 シナリオ

テスト シナリオ

テスト担当者は、コントロールの値が以前に選択されているかどうかに基づいて、**ComboBox** に対して異なる選択を行います。**ComboBox** の既存のテキストが空 (選択されていないことを示す) の場合は **ComboBox** に対する一連の操作をテストし、そうでない場合は別の操作をテストします。

これを行うには、テスト担当者は、操作を実行する前にコントロールの **Text** プロパティにテキストが含まれているかどうかを評価するようにビジュアル テストを設定します。判断ロジックでは、コントロールの **Text** プロパティのテキストの有無に基づいて、後続の操作が決まります。判断ロジックの If/Else 構文に

は、判断条件に基づく異なる操作を含めます。また、テスト担当者は、判断ロジックの結果と実行された操作をテスト結果にレポートする必要があります。

サンプル ビジュアル テスト：コントロールの選択を確認するための判断ロジックの使用：パート 2 詳細

サンプル ビジュアル テスト

以下の図に、このテストシナリオに基づいた **テストステップ** ペインのサンプル ビジュアル テスト ステップを示します。テスト内のステップの進行を強調表示しています。

Test Steps		Actions
1	<<Start>>	
2	Using FormsWindow 'Windows Forms Sample Application'	
3	Select button 'automationId=btnBasicControls'	
4	Using FormsWindow 'Basic Controls'	
5	Select tab 'ToolStrip'	
6	If "ComboBox[2]". "Text" Is Equal to ""	
7	Result Comment: 'ComboBox text is null, selection not made. Select "red" then "green".'	
8	Using FormsWindow 'Basic Controls'	
9	Select combo box item 'red'	
10	Using FormsWindow 'Basic Controls'	
11	Select combo box item 'green'	
12	Else	
13	Result Comment: 'ComboBox NOT null, selection was made. Select "blue" then "black".'	
14	Using FormsWindow 'Basic Controls'	
15	Select combo box item 'blue'	
16	Using FormsWindow 'Basic Controls'	
17	Select combo box item 'black'	
18	End If	
19	<<End>>	

ステップの詳細

ビジュアルテストの各ステップの詳細は、以下のとおりです。

ステップ 1： <<開始>> ステップ。

ステップ 2： 記録された指定の画面を使用して後続の自動ステップが再生されることを示す画面ステップ。このステップが **テストステップ** ペインで強調表示されると、関連する画面が **画面プレビュー** に表示されます。

ステップ 3： テストアプリケーション内の **Button** コントロールをクリックする、記録されたステップ。ボタンをクリックすると、一連のコントロールがさまざまなタブに表示されます。**ToolStrip** タブには、テスト対象のコントロールが含まれています。

ステップ 4 から 5： 記録されたステップは、テスト対象の **ComboBox** コントロールを含む **ToolStrip** タブにアクセスして選択します。

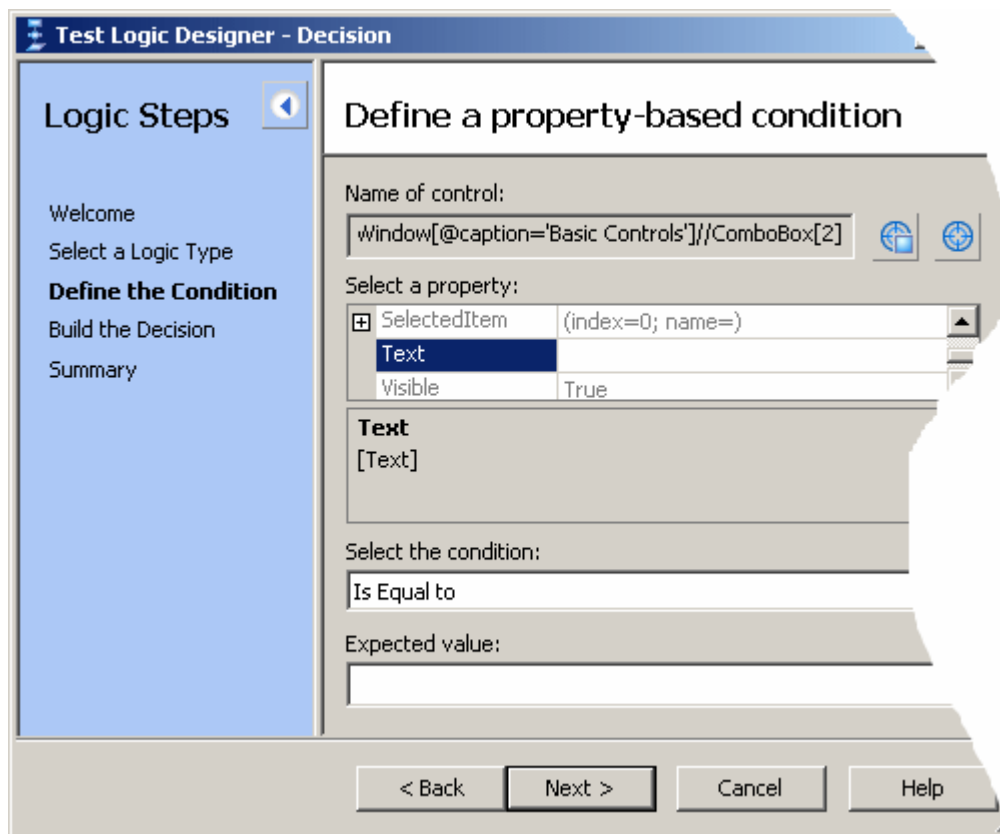
ステップ 6： 判断ロジックを開始する If ステップ。判断ロジックの一部は、テストを記録したあとで、テスト担当者が作成します。ロジック アイコンは、このステップが判断ロジックステップであることを示します。

判断ロジックによって、**ComboBox** コントロールの **Text** プロパティの内容が評価されます。コントロールに値が設定されていない場合、**Text** プロパティは Nothing または空白です。値が Nothing または空白

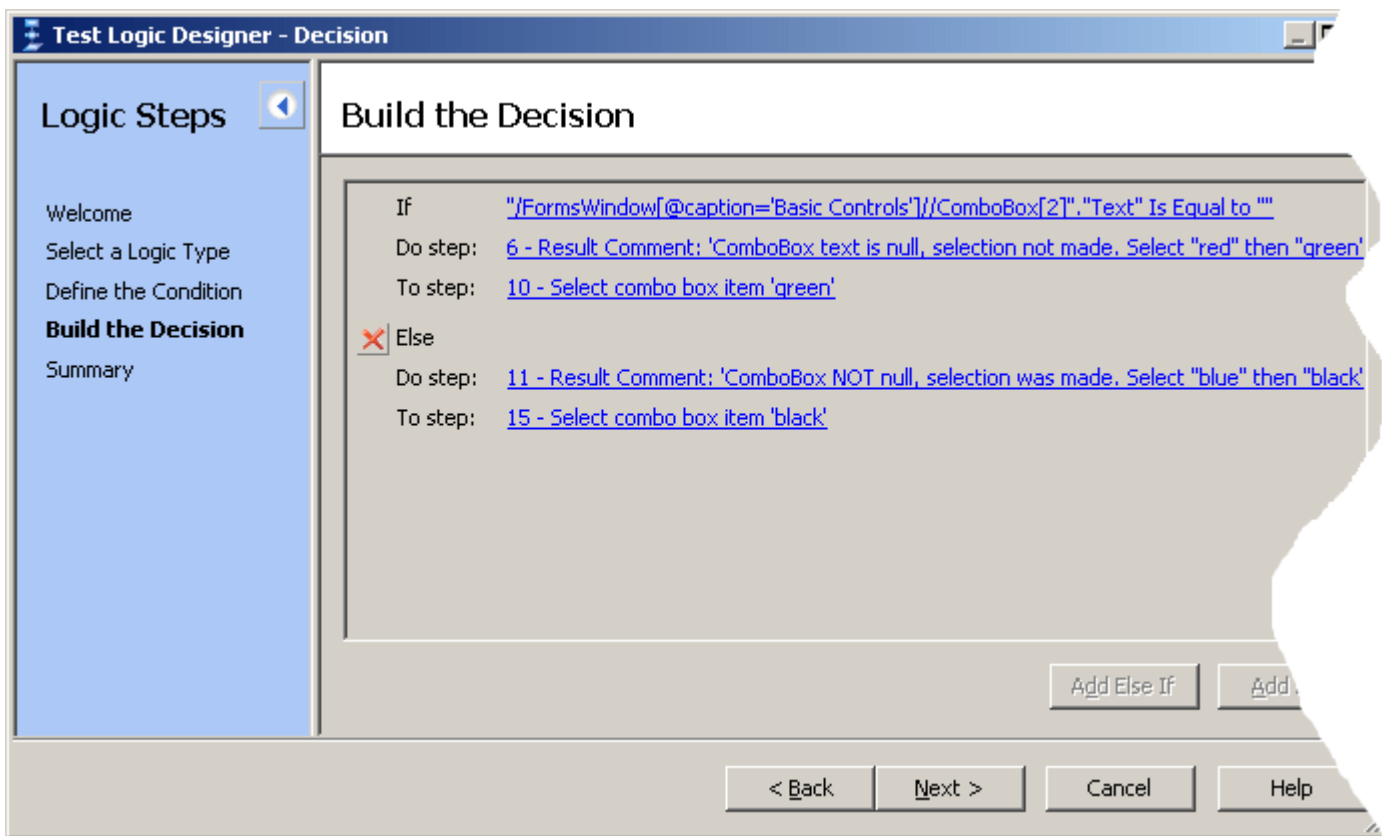
の場合は、ビジュアル テストはある一連のステップを実行し、値が Nothing または空白以外の場合は、異なる一連のステップを実行します。


テスト ロジック デザイナ を使用して判断ロジックを設定するには、以下の操作を実行します。

- **コントロールのプロパティに基づく** ロジックの種類を選択します。
- 以下の操作を行うため、**テスト ロジック デザイナ** の [プロパティに基づく条件の定義] ページを使用します。
 - テスト対象アプリケーションを使用して、コントロールの名前を識別します。
 - **テスト ロジック デザイナ** の **プロパティの選択** リストで **テキスト** を選択します。
 - 以下の図に示すように、条件が **等しい** であり、**期待値** フィールドが空白であることを確認します。



- **テスト ロジック デザイナ** の **判断の構築** ページを使用して、条件ロジックの結果に応じて実行する一連のステップを指定します。このシナリオでは、結果が true の場合にステップ 6 から 10 を実行します。次に、**Else の追加** をクリックして If ステップを判断ロジックに追加し、以下の図に示すように、結果が false の場合にステップ 11 から 15 を実行します。



 **注:** これらのステップがビジュアル テストに挿入されると、If および Else ステップ以降のステップ番号が変わります。このシナリオでは、以下のステップ番号は、前に図に表示されている番号ではなく、判断ロジックがテストに追加されたあとの番号に基づいています。

ステップ 7: テスト担当者によって作成された 結果コメント ステップ。このステップは、判断ロジックが true と評価される場合に実行されます。ステップが実行されると、テスト結果にメッセージが送信されます。このメッセージは、**ComboBox** にテキストがないため、以前に選択されていないことを示します。また、コントロールに使用できる値が後続のステップで選択されることも示します。

ステップ 8 から 11: 判断ロジックが true と評価される場合、これらのステップは、使用可能な異なる **ComboBox** 項目にアクセスして選択します。まず、**ComboBox** の red が選択され、次に、green が選択されます。

ステップ 12: 判断ロジックの Else ステップ。判断ロジックでは、**Else の追加** がロジックに組み込まれている場合に、このステップが作成されます。Else ステップは、判断ロジック内の If ステップの条件に一致しない場合に、判断ロジックの Else ステップと End If ステップの間にあるステップが再生されることを示します。

ステップ 13: テスト担当者によって作成された別の 結果コメント ステップ。このステップでは、テスト結果にメッセージが送信されます。このメッセージは、**ComboBox** テキストが Nothing ではなく、使用可能な項目が選択されることを示します。

ステップ 14 から 17: 判断ロジックが false と評価される場合、これらのステップは、使用可能な異なる **ComboBox** 項目にアクセスして選択します。まず、**ComboBox** の blue が選択され、次に、black が選択されます。

ステップ 18: ビジュアル テストの判断ロジックの一部として作成された End If ステップ。このステップで、If...Else ロジックが閉じます。

ステップ 19: <<終了>> ステップ。ビジュアル テストの再生を完了します。

プロパティ

このトピックでは、ビジュアルテストのプロパティを示します。

アクティブ データ プロパティ

アクティブ データ プロパティ カテゴリには以下のプロパティが含まれます。

保存 オプション 更新されたアクティブ データを指定したアクティブ データ ファイルにすぐに保存するか、または再生中にはまったく保存を行わないかを指定します。
リストから以下のいずれかの値を選択します。

- **アクティブ データ ファイルにデータを直ちに保存する**：ビジュアル テストに関連付けられた指定のアクティブ データ資産で更新されたデータを、ステップ実行時にそれぞれのアクティブ データ ファイルに保存します。ビジュアル テストで使用されるアクティブ データは、通常、式によって更新されます。データを保存するアクティブ データ ファイルを含む資産は、**保存するアクティブ データ** プロパティで指定します。



注：ビジュアル テストに保存操作を取り消すステップが含まれていないかぎり、ビジュアル テストの再生完了時には、ビジュアル テストの再生中に更新されたアクティブ データがそれぞれのアクティブ データ ファイルに保存されます。

- **アクティブ データ ファイルにデータを保存しない**：更新されたアクティブ データは **保存するアクティブ データ** プロパティで指定したアクティブ データ資産のアクティブ データ ファイルに保存されません。

名前 **保存オプション** プロパティで選択した値に応じて、再生中に更新データがまったく保存されないファイルか、またはこのステップの実行時に更新データが保存されたファイルを含むアクティブ データ資産を指定します。詳細については、「アクティブ データ資産のビジュアル テストへの関連付け」を参照してください。

リストから適切なアクティブ データ資産を選択します。リストにはビジュアル テストに関連付けられたアクティブ データ資産が表示されます。

詳細設定プロパティ

詳細設定 プロパティ カテゴリには以下のプロパティが含まれます。

表示する ボタン メッセージ ボックスに表示する 1 つまたは複数のボタンを決定します。
リストから表示するボタンを選択します。リストには以下が含まれています。

- **OK**：メッセージ ボックスに **OK** ボタンを表示します。
- **OK、キャンセル**：メッセージ ボックスに **OK** ボタンと **キャンセル** ボタンを表示します。
- **中止、再試行、無視**：メッセージ ボックスに **中止**、**再試行**、および **無視** ボタンを表示します。
- **はい、いいえ、キャンセル**：メッセージ ボックスに **はい**、**いいえ**、および **キャンセル** ボタンを表示します。
- **はい、いいえ**：メッセージ ボックスに **はい** と **いいえ** ボタンを表示します。
- **再試行、キャンセル**：メッセージ ボックスに **再試行** ボタンと **キャンセル** ボタンを表示します。

デフォルト ボタン メッセージ ボックスを表示するときに自動的に選択するボタンを指定します。
リストから **デフォルト** ボタンを選択します。リストの内容は、**表示するボタン** プロパティで選択した値に基づきます。

表示する アイコン メッセージ ボックスに選択したアイコンを表示します。メッセージの内容を示す適切なアイコンを表示できます。有効な値は以下のとおりです。

- **(なし)** : メッセージボックスにアイコンを表示しません。これはデフォルト値です。
- **重要** : メッセージボックスに **重要** アイコンを表示します。このアイコンを使用して、注意が必要な重大なエラーまたは状態をテスト担当者に警告します。
- **警告の照会** : メッセージボックスに **警告の照会** アイコンを表示します。このアイコンを使用して、応答が必要な状態をテスト担当者に警告します。
- **警告メッセージ** : メッセージボックスに **警告メッセージ** アイコンを表示します。このアイコンを使用して、応答が不要なエラーまたは状態をテスト担当者に警告します。
- **通知メッセージ** : メッセージボックスに **通知メッセージ** アイコンを表示します。メッセージボックスに情報を表示するときにこのアイコンを使用します。

割り当てプロパティ

割り当て プロパティを使用して以下の設定を行います。

- 指定したコントロールの有効なプロパティを変数に割り当てます。これにより、ビジュアルテストでプロパティ値をより効率的に使用できます。これらのプロパティは「コントロールの " プロパティを変数 " に格納します」ステップで表示されます。
- グローバルの取得ステップでグローバル変数の値を格納するローカル変数を指定します。
- グローバルの設定ステップでグローバルパラメータを定義します。
- eCATT スクリプトからインポートするか、または eCATT スクリプトにエクスポートする eCATT スクリプト引数を指定します。



注: このカテゴリに表示されるプロパティはテストステップのタイプに基づきます。このプロパティカテゴリを含むすべてのステップタイプで、すべてのプロパティが使用されるわけではありません。

このカテゴリには以下のプロパティが含まれます。

グローバル変数名 (グローバルの設定ステップで) 作成または (グローバルの取得ステップで) 取得されるグローバル変数の名前を指定します。

目的のグローバル変数名を入力します。グローバル変数名に有効な文字は、大文字と小文字の英数字とアンダースコアです (「ABC」、「abc」、「_」)。グローバル変数名は一意である必要があります。



ヒント: <<開始>> ステップで作成したグローバル変数を取得する場合は、あらかじめその名前を確認しておく必要があります。グローバル変数を設定するビジュアルテストを開いて、適切なグローバルの設定ステップのグローバル変数名プロパティから名前を取得できます。

eCATT 引数名 : eCATT スクリプトからインポートするか、または eCATT スクリプトにエクスポートする eCATT スクリプト引数の名前を指定します。

値をインポートまたはエクスポートする eCATT スクリプト引数の名前を入力します。引数名は、**eCATT 引数コンテナ** ダイアログボックスを使用してビジュアルテストの引数コンテナに定義する必要があります。入力した名前が引数コンテナで定義された名前と一致していない場合、再生エラーが発生します。

プロパティ名 ステップで識別されるコントロールで使用可能なプロパティの名前を指定します。

ステップで識別されるコントロールが、**プロパティ** ウィンドウの **オブジェクトの種類** プロパティに表示されます。リストから使用可能なプロパティを選択して、関連する値を変数に設定します。

ローカル変数名 指定した値が格納されるローカル変数の名前を指定します。

リストからローカル変数を選択します。ローカル変数をリストに表示するには、事前にローカル変数をビジュアルテストで作成しておく必要があります。

コントロールのプロパティをローカル変数に設定するステップの場合、選択したプロパティの値が指定したローカル変数に格納されます。

eCATT スクリプト引数をインポートするステップの場合、スクリプト引数の値が指定したローカル変数に格納されます。

値 開始時にグローバル変数に指定される値。値を入力するか、または、値エリア内を右クリックして、値を割り当てます。

コマンドプロパティ

このカテゴリには以下のプロパティが含まれます。

パス名 起動するアプリケーションまたはプログラムパス名とファイル名を指定します。
ファイルの選択 をクリックし、**ファイルの選択** ダイアログ ボックスを使用して目的のプログラムまたはアプリケーションを起動するファイルに移動し、このファイルを選択します。
実行する操作が **検索** または **探索** の場合、このプロパティではパス名のみが必要です。

操作 アプリケーションまたはプログラムが起動したときに実行する操作を指定します。
実行する操作をリストから選択します。リストには以下が含まれています。

- **開く** (デフォルト) : 指定したパラメータを使用して、指定ファイルを開きます。ファイルには、アプリケーション、ドキュメント ファイル、またはフォルダを指定できます。
- **編集** : エディタを起動して、編集するファイルを開きます。 **パス名** プロパティで指定されたファイルが、ドキュメント エディタで開くことのできるドキュメント ファイルではない場合、この操作は実行されません。
- **探索** : Windows エクスプローラを使用して、 **パス名** プロパティで指定されたフォルダを開きます。
- **検索** : 指定されたディレクトリを起点として、Windows エクスプローラを検索モードで起動します。ディレクトリは、 **パス名** プロパティまたは **ディレクトリ** プロパティを使用して指定できます。 **パス名** プロパティが使用され、かつ、ファイル名が含まれている場合は、テストを再生しても、このステップは実行されません。
- **印刷** : **パス名** プロパティで指定されたドキュメント ファイルを印刷します。 **パス名** プロパティで指定されたファイルがドキュメント ファイルではない場合、テストを再生してもこのステップは実行されません。

パラメータ アプリケーションを起動するために渡すパラメータを指定します。

パス名 プロパティで指定されたファイルがアプリケーション ファイルの場合は、このプロパティを使用して、アプリケーションに渡して起動時に使用するパラメータを含む NULL 終端文字列を指定します。 **パス名** プロパティで指定されたファイルがドキュメント ファイルまたはその他のタイプのファイルの場合は、値を空白のままにします。

ディレクトリ **操作** プロパティで指定された操作に使用できるデフォルト ディレクトリを指定します。

ウィンドウスタイル アプリケーションまたはプログラムが起動したときの表示方法を指定します。

この値は、選択したウィンドウ スタイルを表す整数としてテスト ステップに表示されます。値はリストから選択します。選択できる値は、以下のとおりです。

- **非表示** : プログラムまたはアプリケーションが起動したとき、ウィンドウは非表示です。フォーカスは、他のウィンドウに渡されます。
- **最大化** : アプリケーション ウィンドウを最大化します。
- **最小化** : 起動時にアプリケーション ウィンドウを最小化し、Z オーダーで次に最上位にくるウィンドウをアクティブにします。

- **復元** : アプリケーション ウィンドウを起動して表示します。ウィンドウが最小化または最大化されていた場合は、そのウィンドウの元のサイズと位置に復元されます。この値は、最小化されたウィンドウを復元するときに使用します。
- **表示** : アプリケーション ウィンドウを起動し、現在のサイズと位置に表示します。
- **デフォルトの表示** : アプリケーションを起動したプログラムからの情報に基づいて、ウィンドウの表示状態を設定します。
- **表示の最大化** : アプリケーションを起動し、ウィンドウを最大化して表示します。
- **表示の最小化** : アプリケーションを起動し、ウィンドウを最小化して表示します。
- **表示を最小化し、アクティブ化しない** : 最小化されたウィンドウでアプリケーションを表示します。現在のアクティブ ウィンドウは、アクティブなままです。
- **現在の状態で表示し、アクティブ化しない** : 現在のウィンドウ状態でアプリケーションを表示します。アクティブ ウィンドウは、アクティブなままです。
- **最新の位置に表示し、アクティブ化しない** : 最後に使用したときのウィンドウ サイズと位置でアプリケーションを表示します。アクティブ ウィンドウは、アクティブなままです。
- **通常表示** : アプリケーションを起動して、ウィンドウを表示します。最後に使用したウィンドウの位置と最小化または最大化の状態が復元されます。この値は、はじめてウィンドウを表示するときに使用します。

遅延プロパティ

このカテゴリには以下のプロパティが含まれます。

遅延タイプ ステップの遅延間隔を設定します。

プ リストから **秒** または **ミリ秒** を選択します。

遅延時間 遅延タイプに基づいて遅延時間を指定します。

リテラル値を入力するか、あるいは **選択** をクリックして変数、式の結果、またはアクティブデータ値を割り当てます。

フラグの設定プロパティ

フラグの設定 プロパティ カテゴリには以下のプロパティが含まれます。

エラー時のフラグの作成 エラー発生時にフラグを作成するかどうかを指定します。このプロパティを編集するには、値を **True** または **False** に設定します。 **True** に設定した場合、フラグが作成されます。 **False** に設定した場合、フラグは作成されません。

フラグの説明 エラー時に作成されるフラグの説明を含めます。説明は、ビジュアル テストの結果と **開始画面** に表示されます。このプロパティを編集するには、フラグの説明を入力します。

全般プロパティ

全般 プロパティ カテゴリには以下のプロパティが含まれます。

メッセージボックスのタイトル メッセージ ボックスのタイトル バーに表示するテキストを決定します。デフォルト値は、Silk Test です。

デフォルトを変更するには、値を入力するか、**選択** をクリックして、メッセージ ボックスのタイトルとして値を割り当てます。

メッセージテキスト メッセージ ボックスに表示されるテキストの内容を指定します。事実上、ビジュアル テストで使用可能ななどの情報でもメッセージの内容に使用できます。

メッセージの内容を直接入力するか、**選択** をクリックし、メッセージ テキストとして値を割り当てます。

リテラルテキストと変数に返されたデータなどの情報を組み合わせるには、**選択** をクリックしてメニューから **式** を選択します。**式デザイナー** を使用して、文字列の組み合わせを作成します。

たとえば、説明テキストと ComboBox の項目数を表示する場合、以下のような式になります。

```
"Message box text is: " + ToString (intCount)
```

ここで、各値の内容は以下のようになります。

"Message box text is: " は説明テキストの文字列です。

+ は文字列の組み合わせに使用する演算子です。

ToString (intCount) は ToString () 変換関数を使用して文字列に変換される、コントロール intCount 内の項目数を保持する変数です。

アクティブデータ データ ファイルがビジュアル テストに関連付けられているアクティブ データ資産の数を表示します。

値エリアで **アクティブデータの参照** をクリックし、**アクティブデータの参照** ダイアログ ボックスを使用して資産を選択することによって、アクティブ データ資産をビジュアル テストに関連付けることができます。

項目 n ビジュアル テストに関連付けられているアクティブ データ資産の名前を表示します。

再生設定の取得プロパティ

再生設定の取得 プロパティ カテゴリには以下のプロパティが含まれます。

再生設定 個々の再生情報を制御します。

「再生設定 " を取得して、ローカル変数 " に格納します」ステップでは、値は再生設定で、この再生設定の現在の設定値がローカル変数に格納されます。値は以下のいずれかから取得されます。

- 再生オプションに設定されている再生設定の現在値。
- 定義済みの実行環境（ビジュアル テストで **RunEnvironment** 再生設定が設定されている場合）。
- ビジュアル テストの **再生設定の設定** ステップ。

リストから **再生設定** を選択します。

ローカル変数名 値が格納されるローカル変数の名前を指定します。

「再生設定 " を取得して、ローカル変数 " に格納します」ステップでは、選択した再生設定の値が指定したローカル変数に格納されます。リストからローカル変数を選択します。ローカル変数をリストに表示するには、事前にローカル変数をビジュアル テストで作成しておく必要があります。

その他のプロパティ

その他 プロパティ カテゴリには以下のプロパティが含まれます。

説明 選択したテスト ステップの説明を表示します。このプロパティを編集するには、プロパティ値 テキスト ボックスに選択したテスト ステップの説明を入力します。

最新の変更 テスト ステップが最後に変更された日付と時刻を表示します。このプロパティは読み取り専用です。

オブジェクト プロパティ

オブジェクト プロパティ カテゴリには以下のプロパティが含まれます。

- クラス名** 選択されたオブジェクトのクラス名を識別します。
- ロケータ** テストするオブジェクトを識別します。 **画面プレビュー** ペインまたは実際のアプリケーションを使用すると、テストするオブジェクトを識別できます。あるいは、リテラル値、式、アクティブ データ、または変数を選択し、ロケータとして使用できます。
- プロパティ** 指定したオブジェクトから取得できるプロパティ、および変数への割り当てが可能なプロパティのリストを返します。リストからプロパティを選択します。あるいは、リテラル値、式、アクティブ データ、または変数を選択し、プロパティとして使用できます。
- ローカル変数名** 指定した値が格納されるローカル変数の名前を指定します。

エラー時の移動先プロパティ

エラー時の移動先 プロパティ カテゴリには以下のプロパティが含まれます。

実行する操作 ビジュアル テストの再生中にエラーが発生した場合に、実行する操作を設定します。リストから以下のいずれかのオプションを選択します。

- **デフォルトの再生エラー ダイアログを表示します : 再生エラー** ダイアログ ボックスを表示し、そこでそのエラーに対して実行する操作を決定します。 **再生エラー** ダイアログ ボックスを使用してデバッグ モードに入り、ビジュアル テストでエラーを診断します。
再生エラー ダイアログ ボックスの表示は、再生でエラーが発生した場合のデフォルトの動作です。
- **再生の終了** : エラーが発生したときに、ビジュアル テストの再生を終了します。ビジュアル テストのその他のステップは再生されません。再生が終了すると、 **再生完了** ダイアログ ボックスが表示され、そこで再生の完了時に実行する操作を決定します。
- **テストの終了 : 資産** プロパティに指定されているビジュアル テストを再生してから、エラーが発生したステップで終了します。再生が終了すると、 **再生完了** ダイアログ ボックスが表示され、そこで再生の完了時に実行する操作を決定します。
- **ステップの再試行 : 資産** プロパティに指定されているビジュアル テストを再生してから、エラーが発生したステップの実行を試行します。 **資産** プロパティにビジュアル テストが指定されていない場合、単にエラーが発生したステップの再実行を試行します。 **再試行の回数** プロパティには、再生を試行する回数を指定できます。デフォルト値は、5 です。指定した再試行の回数に到達した時点でステップ実行に失敗すると、再生エラーが生成されます。



注: 再生エラーは、ステップを再実行している間は Silk Test Workbench は生成しません。 **再試行の回数** プロパティに 0 を設定すると、ステップは永久に再試行されます。これにより、再生時に失敗したステップが継続して再試行されている間、Silk Test Workbench が応答しなくなったように見える場合があります。

この **ステップの再試行** 操作は、ビジュアル テストで、エラーを確実に処理したり、テストで問題が予想される潜在的なエラーを修正したりするために役立ちます。たとえば、ビジュアル テストでアクセスするブラウザ アプリケーションのページが、確実に読み込まれていない可能性がある場合、「アタッチの失敗」エラーが発生することがあります。 **ステップの再試行** 操作では、ビジュアル テストを、ページを確実に読み込んでからそのページとの対話を試行する別のビジュアル テストに移行させることによって、再生エラーを解決できます。

- **再開して次に進む : 資産** プロパティに指定されているビジュアル テストを再生してから、エラーが発生したステップの直後のステップの実行を試みます。エラーの原因となったステップは、再実行されません。
- **移動 : 資産** プロパティに指定されているビジュアル テストを再生し、 **ラベル** プロパティに指定されている事前定義のラベル ステップに移動して、再生を続行します。 **移動** を使用すると、ビ

ビジュアルテストで、再生中に発生したエラーと同様のエラーが発生する可能性のあるステップをスキップして、テストの別のセクションに移動できます。



注: エラー処理に使用された特定のラベルステップが、エラーが発生するステップの前にある場合、関係するステップがループして繰り返し再生されるため、そのエラーは繰り返し発生します。

資産 ビジュアルテストの再生中にエラーが発生した場合に実行するビジュアルテストを指定します。別のビジュアルテストを選択するには、**ビジュアルテストの参照** をクリックしてから、ビジュアルテストを選択します。



ヒント: 資産を指定してステップを作成したあとに **実行する操作** プロパティで別の操作を選択すると、ステップの動作を変更できます。たとえば、**実行する操作** プロパティを **テストの終了** に設定し、**資産** プロパティで資産を選択した場合、ステップテキストは次のようになります：

エラー時にビジュアル テスト '[ビジュアル テスト]' を再生して 'テストの終了'

ここで **実行する操作** プロパティの値を変更すると、ステップの動作を変更できます。値を **ステップの再試行** に変更すると、ステップは次のようになります：

エラー時にビジュアル テスト '[ビジュアル テスト]' を再生して 'ステップの再試行'

ラベル エラーが発生した場合にビジュアルテストが移動できる使用可能なラベルステップのリストが含まれます。リストからラベルステップの名前を選択します。

このプロパティは、**実行する操作** プロパティが **移動** に設定されている場合にのみ、表示されます。

ビジュアルテストの再生設定

ビジュアルテストの再生動作の構成には、以下の再生設定を使用できます。

アプリケーション準備完了タイムアウト 新しく起動したアプリケーションが準備完了状態になるまで待機する時間をミリ秒で指定します。指定したタイムアウト時間内にアプリケーションの準備が整わない場合は、Silk Test Workbench によって例外がスローされます。

ウィンドウを閉じるためのボタン CloseSynchron メソッドでウィンドウを閉じるために使用するボタンを指定します。

確認ダイアログに対するボタン CloseSynchron メソッドでウィンドウを閉じる際に表示された確認ダイアログ ボックスを閉じるために使用するボタンを指定します。

無応答のアプリケーションを閉じる 無応答のアプリケーションを閉じるかどうかを指定します。タイムアウトなどの理由で、エージェントとアプリケーション間の通信に失敗した場合に、アプリケーションが無応答になります。

ウィンドウを閉じるためのタイムアウト ウィンドウ閉じるための次の方式を試行する前に待機する時間をミリ秒で指定します。最終的に失敗と判断する前に、4 種類の方式が Agent によって実行されます。つまり、閉じるのに失敗するまでにかかる合計時間は、指定した値の 4 倍の時間になります。

コントロールのキャプチャ 再生時に Silk Test Workbench がコントロールとそのプロパティをキャプチャできるようにします。記録オプションでコントロールのキャプチャ設定をいいえに設定した場合は、このオプションが上書きされ、再生時にコントロールがキャプチャされます。



注: 記録の **画面キャプチャ** 設定を **なし** に設定した場合は、**コントロールのキャプチャ** を **はい**、再生の **画面キャプチャ** 設定を **なし** 以外の値に設定して、再生時にコントロールをキャプチャする必要があります。

オブジェクトがアクティブであることを保証する	ターゲット オブジェクトがアクティブであることを保証します。
再生中にオブジェクトをハイライトする	現在のオブジェクトが再生中にハイライトされるかどうかを指定します。
キーボード イベント遅延	再生時のキー入力間の遅延をミリ秒で指定します。 テストするアプリケーションに応じて、選択する最適な値が異なることに注意してください。たとえば、Web アプリケーションをテストする場合、1 ミリ秒に設定すると、ブラウザが極端に遅くなります。ただし、この値を 0 (ゼロ) に設定すると、基本的なアプリケーション テストに失敗する可能性があります。
ダイアログを閉じるためのキー	CloseSynchron メソッドでウィンドウを閉じる際に表示されたダイアログ ボックスを閉じるキー シーケンスを指定します。例: 「<ESC>」、 「<Alt+F4>」
ウィンドウを閉じるためのメニュー項目	CloseSynchron メソッドでウィンドウを閉じるために使用するメニュー項目を指定します。例: 「ファイル/*終了」、 「ファイル/*中止」
マウス イベント遅延	各マウス イベントの前に使用される遅延をミリ秒で指定します。
オブジェクト有効化タイムアウト	再生中にオブジェクトが有効になるまで待機する時間をミリ秒で指定します。オブジェクトが有効になるとすぐに、Silk Test Workbench はこのオブジェクトと対話することができます。たとえば、オブジェクトがボタンの場合、このボタンが有効になると、Silk Test Workbench でボタンをクリックできるようになります。
オブジェクト解決タイムアウト	再生中にオブジェクトの解決を待機する時間をミリ秒で指定します。オブジェクトが解決されるとすぐに、Silk Test Workbench はこのオブジェクトを認識することができます。
再生モード	コントロールの再生方法を定義します。「低レベル」を使用すると、マウスとキーボードを使用して各コントロールが再生されます。「高レベル」を使用すると、API を使用して各コントロールが再生されます。コントロールごとにデフォルトの再生モードが割り当てられています。デフォルトの再生モードが選択されると、各コントロールごとのデフォルトの再生モードが使用されます。デフォルト モードを使用すると、最も信頼できる結果が得られます。「低レベル」または「高レベル」の再生を選択すると、すべてのコントロールの再生モードが選択した再生モードで上書きされます。
結果合格条件 (パーセンテージ)	今後の実行の成功を定義する全般条件を設定します。たとえば、結果の合格条件を 90% に設定した場合、10 件の検証のうち 9 件に合格しないと、再生の結果が合格になりません。
すべての情報を保存	再生が完了したあとに、ビジュアル テストのすべての情報を保存するかどうかを指定します。
画面キャプチャ	Silk Test Workbench で再生中にテスト アプリケーションの画面をキャプチャする方法を指定します。 以下のオプションがあります。 <ul style="list-style-type: none"> • 記録と同じ: ビジュアル テストが記録されたときに 記録 オプションに存在した 画面キャプチャ 設定と同じ設定を使用します。 • アクティブ ウィンドウ: 記録中にテスト アプリケーションのアクティブなウィンドウのみをキャプチャします。 • アプリケーション: テスト アプリケーションとテスト アプリケーション内のすべてのウィンドウをキャプチャします。

- **デスクトップ** : テスト アプリケーションを含む、記録中にデスクトップに表示されているすべてのアプリケーションの画面をキャプチャします。
- **なし** : 記録中に画面をキャプチャしません。



注: この設定を **なし** に設定すると、**コントロールのキャプチャ** および **画面キャプチャ (テスト ステップ)** 設定に指定した値に関係なく、コントロールまたは画面は **結果** ウィンドウにキャプチャされません。

画面キャプチャ (テスト ステップ)

Silk Test Workbench が記録中にテスト アプリケーションの個別の画面をキャプチャするかどうかを決定します。これによって、**テスト ステップ** ペインの個々のテスト ステップと連動して、**画面プレビュー** の個々の画面で発生する操作を表示できるようになります。また、再生時に記録される画面を、ビジュアル テストを最初に記録したときにキャプチャされた画面と比較できます (記録時にもテスト ステップの画面がキャプチャされた場合)。

以下のオプションがあります。

- **記録と同じ** : ビジュアル テストが記録されたときに **記録** オプションに存在した **画面キャプチャ (テスト ステップ)** 設定と同じ設定を使用します。
- **はい** : 記録中に画面をキャプチャします。 **画面キャプチャ** 設定を使用して、画面をキャプチャする方法を決定します。
- **いいえ** : 再生中に個別のテスト ステップをキャプチャしません。 **画面キャプチャ** 設定が **なし** 以外に設定された場合、個別のテスト ステップに関連付けられたグループ画面が **画面プレビュー** に表示されます。

スクリーンショット遅延

新しいウィンドウにフォーカスを設定してから、ウィンドウのスクリーン ショットを取得するまで待機する時間をミリ秒で入力します。

同期除外リスト

ページ同期中に除外するサービスまたは Web ページの URL を指定します。AJAX フレームワークやブラウザによっては、サーバーから非同期にデータを取得するために、特殊な HTTP 要求を継続して出し続けるものがあります。これらの要求により、指定した同期タイムアウトの期限が切れるまで同期がハングすることがあります。この状態を回避するには、HTML 同期モードを使用するか、問題が発生する要求の URL を **同期除外リスト** 設定で指定します。

複数の URL をカンマで区切って指定できます。 `http://test.com/timeService` のように URL 全体を入力したり、 `timeService` のように URL の一部を入力します。

同期モード

サポートする同期モード (HTML または AJAX) を設定します。HTML モードを使用すると、すべての HTML ドキュメントが対話的な状態になることが保証されます。このモードでは、単純な Web ページをテストすることができます。Java Script が含まれる、より複雑なシナリオが使用される場合は、同期関数を手動で記述することが必要になる可能性があります。AJAX モードを使用すると、同期関数を手動で記述する必要はなくなります。

同期タイムアウト

オブジェクトが準備完了状態になる最大時間をミリ秒で指定します。

無応答のアプリケーションのタイムアウト

応答のない再生操作をキャンセルするまでのタイムアウト値をミリ秒で指定します。

挿入されたスクリプトまたはビジュアル テストの再生設定

再生設定 は、ビジュアル テスト内で実行されているビジュアル テストまたはスクリプトで使用される再生設定を指定します。リストから以下のいずれかの値を選択します。

システムのデフォルトを使用	各再生オプションについて、 オプション ダイアログ ボックスに表示されているデフォルトを使用します。
親の設定を継承する	現在の再生の実行に、親ビジュアル テストの再生設定を使用します。
親から設定をコピー	現在の再生の実行および今後すべての再生の実行に、親ビジュアル テストの再生設定を使用します。

結果プロパティ

結果 プロパティ カテゴリには以下のプロパティが含まれます。

ステップ	テスト ステップ ペインで選択したテスト ステップの名前を表示します。
結果	テスト ステップ ペインで選択したステップの結果を表示します。
結果の詳細	テスト ステップ ペインで選択したステップの結果の詳細を表示します。
行番号	テスト ステップ ペインで選択したステップの行番号を表示します。
コマンドの詳細	テスト ステップ ペインで選択したステップのログの詳細を表示します。
日付/時刻	テスト ステップ ペインで選択したステップの日付と時刻を表示します。
ミリ秒	テスト ステップ ペインで選択したステップのミリ秒数を表示します。
名前	テスト ステップ ペインで選択したステップの資産名を表示します。
ユーザー名	ビジュアル テストを再生したユーザーの名前を表示します。
マシン名	ビジュアル テストが再生されたコンピュータの名前を表示します。

再生設定の設定プロパティ

再生設定の設定 プロパティ カテゴリには以下のプロパティが含まれます。

再生設定	リストから 再生設定 を選択します。
値	<p>選択した再生設定に値を割り当てます。</p> <p>割り当てられた値は、再生設定の設定 ステップを含むビジュアル テストでのみ使用されます。</p> <p>選択した再生設定に従って、リストから値を選択するか、選択 をクリックし、再生設定値として値を割り当てます。</p>

タイマープロパティ

タイマー プロパティ カテゴリには以下のプロパティが含まれます。

タイマーの動作	<p>タイマー テスト ステップの操作を設定します。</p> <ul style="list-style-type: none"> • 開始：タイマーを開始します。タイマーが停止すると、タイマーは 0 にリセットされます。すでに実行されているタイマーを開始しようとすると、エラーが発生します。 • 停止：タイマーを停止します。タイマーには時間が保持され、必要な場合は 再開 操作を使用してタイマーを再開できます。 • 再開：停止 操作を使用して停止されたタイマーを再開します。タイマーは以前に記録された時間からカウントを再開します。 • リセット：時間を 0 にリセットします。
---------	---

タイマー番号 タイマーの番号を設定します。複数の条件の時間を計るときに、各タイマーに一意的な番号を割り当ててタイマーを区別できます。リストからタイマーの番号を選択します。ビジュアル テスト

には、最大で 10 個のタイマーを組み込むことができます。このプロパティは、一連のステップの実行にかかる時間を検証する場合のみ有効です。

検証結果の説明プロパティ

検証結果の説明プロパティ カテゴリには以下のプロパティが含まれます。

- 検証の '合格 テキスト'** 指定した検証ロジックの条件に適合した場合にビジュアル テストの結果説明に表示するメッセージが含まれます。このプロパティを編集するには、プロパティを選択して、必要な説明を入力します。
- 検証の '失敗 テキスト'** 指定した検証ロジックの条件に適合しなかった場合にビジュアル テストの結果説明に表示するメッセージを作成します。このプロパティを編集するには、プロパティを選択して、必要な説明を入力します。

待機プロパティ

待機プロパティ カテゴリには以下のプロパティが含まれます。

- ロケータ** ステップが、どのオブジェクトが存在または消滅するまで待機するかを指定します。以下のいずれかの方法を使用して、コントロールを識別します。
- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別ダイアログ** ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
 - **選択**：このボタンをクリックして、リテラル、変数、式の結果、またはアクティブ データ値を割り当てます。
- 待機の種類** オブジェクトが存在、または消滅するまで待機する時間を指定します。
- タイムアウト** オブジェクトが存在、または消滅するまで待機する最大時間をミリ秒で指定します。このプロパティを編集するには、プロパティを選択して数値を入力します。デフォルト値は、**デフォルトの待機タイムアウト (ミリ秒)** オプション値に設定されています。タイムアウト期間内に一致するオブジェクトがない場合は、再生エラーが発生します。

スクリプトの作成

Silk Test Workbench は、ビジュアル テストと同じようにスクリプトを使用して、アプリケーションのテスト中に実行する操作を模倣します。ユーザーがキー入力やマウス操作でメニュー、リスト項目、ボタンなどを選択するのと同じように、アプリケーションや Web ページを制御します。記録中、Silk Test Workbench によりアプリケーションをテストするキー入力やマウス クリックがすべて生成されます。

Silk Test Workbench のスクリプト言語は、Microsoft .NET Framework 環境で実行されるすべてのアプリケーションを完全に制御できる堅牢なプログラミング言語である、Microsoft の Visual Basic です。.NET スクリプトは、高水準プログラミング言語が持つ諸機能の他、ソフトウェアの制御とテスト専用に設計された機能も備えています。この言語を使用すると、以下のようなスクリプトを作成できます。

- 指定した時刻に自動的に実行する。
- 完全に無人で実行する。
- ユーザーとの対話を行って、ID やパスワードなどの認証情報を受け取ることができる。

メニューの選択、データの入力、処理の確認など、アプリケーションのテストに使用する操作は、スクリプトで VB.NET コマンドとして表されます。これらのコマンドはスクリプトに挿入され、変更や再生が可能です。

Silk Test Workbench により、スクリプトをすばやく記録し実行することができます。スクリプトには、記録できない「ハンドコーディング」のステートメントを追加したり、テストアプリケーションの変更を反映した修正を加えたりできます。また、既存のスクリプトからコードをカットアンドペーストして新しいスクリプトを作成することもできます。また、**オブジェクトの識別** ダイアログ ボックスを使用してロケータまたは個々のオブジェクトのオブジェクト マップ項目を記録し、それをスクリプトで使用することもできます。

スクリプトをすばやく作成するには、記録機能を使用するのが最適の方法です。記録機能により、Silk Test Workbench が生成するスクリプトのコーディング方法を学習することができます。これによりスクリプトの作成の基本を理解し、さらに効率的なテスト ソリューションの作成、スクリプトのモジュール化、独立したコード モジュールの作成についての理解を深めることができます。

スクリプトを使用する利点

スクリプトでは、アプリケーションをテストするために、操作がオブジェクトレベルで記録されます。これらの操作には、メニューの選択、データの入力、アイコンのクリック、結果の確認などが含まれます。テスト手順をスクリプトで管理すると、以下の利点があります。

- スクリプトは高度に拡張可能であるため、機能と柔軟性を強化できます。.NET スクリプトの機能を使用して、複雑なテスト ケースに対する複雑なソリューションを開発できます。
- スクリプトでは、外部アセンブリで提供される機能を活用して、スクリプト内で使用できる機能を拡張できます。
- 開発者とテスト担当者が、共通スクリプト言語 VB.NET を共有できます。
- テストアプリケーションが変更された場合に、スクリプトを最初から作成し直す必要がありません。
- 既存のスクリプトをコピーして変更することにより、新しいテスト手順を簡単に作成できます。
- テストへのモジュール式アプローチが推進され、再利用率を高めることができます。他のスクリプトまたは機能呼び出すことができます。
- スクリプトをループさせて、プロセスを繰り返しテストできます。
- 予想結果を検証し、予想しない状況が発生した場合にそれに対処する、インテリジェントなテストを迅速に作成できます。
- テストプロセスを文書化できます。
- テストアプリケーションが完了する前に、スクリプトを作成できます。

テスト スクリプト作成のベスト プラクティス

テスト ケースの書き方によっては、テスト セットのパフォーマンスと安定性に非常に大きな影響を与える場合があります。記録時に、可能な限り速く安定したスクリプトを Silk Test Workbench は作成します。しかし、テスト スクリプトを手動で作成したり編集する必要がある場合もあります。このトピックでは、いくつかの一般的なガイドラインを提示します。このガイドラインに従うことによって、保守性、再利用性の高いテスト スクリプトの作成、およびテストの安定性が高まるでしょう。

- テストの名前の一貫性を保ち、内容のわかりやすい名前をつけてください。テスト対象アプリケーションとテストの機能を表す名前にすると良いでしょう。たとえば、テストの名前を *MyApp_SuccessfulLogin* や *MyApp_FailingLogin* とすると、*Untitled_42* や *Untitled_43* とするよりも、他のユーザーにとって非常にわかりやすいものになります。
- できるだけ細かくテスト ケースの説明をコメントに記述してください。テスト ケースについての説明が普通の言葉で記述されていないと、実装されたコードを変更する必要がある人が、そのテストが正確に何を行っているのかを理解することが困難になります。
- テスト ケースを開始する時に、テスト対象アプリケーションが確実に適切な状態になるようにしてください。テスト ケースの操作を実行する前に、テスト対象アプリケーションを正しい状態に戻します。

- テストケースを完了する時に、テスト対象アプリケーションが確実に適切な状態になるようにしてください。追加のテストがそのテストの結果に依存する場合は、それらが開始できる状態にします。テストケースの操作が実行される時に、テスト対象アプリケーションを正しい状態に戻します。
- 可能な限り、テストケースは他のテストケースの結果に依存しないようにしてください。これが不可能な場合、テストケースが正しい順番で実行されるようにしてください。
- 機能的な流れだけでなく、テスト対象アプリケーションの正確さをテストするために、テストに検証を追加してください。
- キーワード駆動テストを使用して、操作セットの再利用性を高めてください。共通に使用する操作をキーワードにまとめ、頻繁に同じ順番で実行されるキーワードをキーワードシーケンスに結合し、キーワードやキーワードシーケンスの組み合わせをキーワード駆動テストとして実行します。
- テストの保守性と再利用性を保つためには、複雑なテストケースを記述するよりも、複数の単純なテストケースを記述して組み合わせる方が好まれます。
- テストセットの冗長性を避けるためには、新しいテストケースを追加するよりも、既存のテストケースを更新することが好まれます。

スクリプトの構文

記録されたスクリプトの行の構文を理解すると、スクリプトが読みやすくなります。オブジェクトをテストするコマンドは、同一の構文要素で構成されています。構文要素には、アプリケーション ウィンドウ (アプリケーションのメイン ウィンドウやダイアログ ボックスなど) のための With ステートメントに続き、クラス、属性、操作を識別するロケーターまたはオブジェクト マップ項目が含まれます。

オブジェクト マップの使用

オブジェクト マップはテスト資産の一種であり、コントロールまたはウィンドウのロケーターではなく、コントロールまたはウィンドウに論理名 (エイリアス) を関連付ける項目が含まれています。

Silk Test Workbench では、スクリプトを記録すると、デフォルトで、スクリプト コンテキストにオブジェクト マップ項目が組み込まれます。

以下の例は、Web アプリケーションをテストするスクリプトで記録される一般的な操作を示しています。

```
With _desktop.BrowserApplication("webBrowser")
  With .BrowserWindow("browserWindow")
    .DomListBox("quickLinkJumpMenu").Select("Auto Quote")
  End With
End With
```

With_ desktop.<application> の部分は、メイン アプリケーション ウィンドウを示しています。

上記の例の DomListBox の部分は、使用するクラスを示しています。

("quickLinkJumpMenu") の部分は、オブジェクトの属性を示しています。この場合、属性はリスト ボックスのリンク メニューを示しています。一方、Select() の部分は、オブジェクトに対して実行する操作またはコマンドを示しています。

最初の With コマンドが定義されたら、そのウィンドウに対する追加の呼び出しでコマンドを繰り返す必要はありません。テストする追加の各ウィンドウに対して、With ステートメントを指定する必要があります。たとえば、以下のコードは、同じスクリプト内から複数のウィンドウを呼び出す方法を示しています。

```
With _desktop.Window("untitledNotepad")
  .MenuItem("aboutNotepad").Select()
  With .Dialog("aboutNotepadDialog")
    .PushButton("ok").Select()
  End With
End With
```

オブジェクト マップ項目はかっこ引用符 ("") で囲まれ、ロケーターのキャプションを使用する必要がなくなります。

ローケータの使用

Silk Test Workbench では、識別されたオブジェクトのリテラル参照はローケータと呼ばれます。Silk Test Workbench では、スクリプトを記録すると、デフォルトで、スクリプト コンテキストにオブジェクト マップ項目が組み込まれます。オブジェクト マップをオフにすると、スクリプトの記録時に、オブジェクト マップ項目ではなくスクリプト コンテキストにローケータが含まれます。

以下の例は、Web アプリケーションをテストするスクリプトで記録される一般的な操作を示しています。

```
With _desktop.BrowserWindow("/BrowserApplication[1]//BrowserWindow")
    .DomLink("@textContents='Court: Gender pay lawsuit can go to trial']").Select()
End With
```

With_ desktop.<application> の部分は、メイン アプリケーション ウィンドウを示しています。

最初の例の DomLink の部分と 2 番めの例の MenuItem の部分は、使用するクラスを示しています。

("@textContents='Court: Gender pay lawsuit can go to trial'") の部分は、オブジェクトの属性を示しています。この場合、属性はテキスト コンテンツを示しています。一方、Select() の部分は、オブジェクトに対して実行する操作またはコマンドを示しています。

最初の With コマンドが定義されたら、そのウィンドウに対する追加のローケータ文字列でコマンドを繰り返す必要はありません。テストする追加の各ウィンドウに対して、With ステートメントを指定する必要があります。たとえば、以下のコードは、同じスクリプト内から複数のウィンドウを呼び出す方法を示しています。

```
With _desktop.Window("@caption='Untitled - Notepad'")
    .MenuItem("@caption='About Notepad'").Select()
    With .Dialog("@caption='About Notepad'")
        .PushButton("@caption='OK'").Select()
    End With
End With
```

caption 属性は、テストするオブジェクトを示しています。caption は、アプリケーションのメイン ウィンドウ ("@caption='Untitled - Notepad'") に続き、メニュー項目 ("@caption='About Notepad'")、後続のダイアログ ボックス ("@caption='About Notepad'")、クリックするボタン ("@caption='OK'") を示しています。

ドライバ スクリプト

ドライバ スクリプトは、テスト アプリケーションをテストするための別のスクリプトをトリガし、それを監視します。ドライバ スクリプトでは、条件ロジックまたは他の制御ロジックを使用してドライバ内の他のテストのオプションの実行を管理することはありますが、通常、ドライバ スクリプト自体はテストを実行しません。

1 つのテストスイートには、テスト アプリケーション内の異なる時点で実行される、多数のテストが含まれています。テストスイートにある各スクリプトは、ドライバ スクリプトによって呼び出すことができます。ドライバ スクリプトは、スクリプトと同様に作成しますが、テスト アプリケーションの実際のテストは行いません。テスト アプリケーションをテスト開始の正しいポイントまで「ドライブ」(駆動) し、1 つまたは複数のスクリプトを呼び出してそのテスト サイトでテストを開始するだけです。テストが終了すると、ドライバ スクリプトに制御が戻され、次のテスト サイトに進みます。ドライバ スクリプトの作成は、個々のスクリプトを手動で記録する場合と同様に行うことができます。

ドライバ スクリプトは、アプリケーションのテストに対するモジュール式アプローチを推進するものです。完全なテストを実行する 1 つのスクリプトを作成する代わりに、テスト フロー全体のうちの個々の部分をテストする再利用可能なスクリプトを複数作成して、ドライバ スクリプトで各部分呼び出すようにすることができます。テスト全体のうちの一部を変更する場合は、変更が必要なスクリプトだけを更新します。ドライバ スクリプトを変更する必要はありません。

適切に設計されたスクリプトは、完了前にテスト アプリケーションを元のテスト サイトに戻します。これには以下のような目的があります。

- ドライバ スクリプトが常に停止した箇所から再開できるようにする。
- ドライバのパスを変更せずに、テスト サイトでテストを追加したり削除したりできるようにする。

ドライバ スクリプトの作成

ドライバ スクリプトは、アプリケーションのテストに対するモジュール式アプローチを推進するものです。完全なテストを実行する 1 つのスクリプトを作成する代わりに、テスト フロー全体のうちの個々の部分をテストする再利用可能なスクリプトを複数作成して、ドライバ スクリプトで各部分呼び出すようにすることができます。

1. 新しいスクリプトを記録します。
たとえば、このスクリプトは、テスト アプリケーションを起動したり、ダイアログ ボックスまたはウィンドウを表示したりします。Web アプリケーションをテストしている場合は、ドライバ スクリプトはブラウザを起動したり、特定の Web ページを表示したりします。
2. 別のスクリプトを再生する箇所まで到達したら、スクリプトの記録を停止します。
3. コード ウィンドウで、カーソルを新しい行に移動し、以下のように入力します。

```
Workbench.RunScript ("script")
```

script はスクリプト名です。

適切に設計されたスクリプトは、完了前にテスト アプリケーションを元のテスト サイトに戻します。これは、ドライバ スクリプトが常に停止した箇所から再開できるようにするためです。

Oracle データベースに格納されるスクリプト名では、大文字と小文字が区別されます。Oracle データベースから挿入するスクリプトの名前は、データベースでの名前と正確に一致する必要があります。一致していないと、Silk Test Workbench でスクリプトの読み込みに失敗します。


4. テスト アプリケーション内の別のテスト サイトでスクリプトを再生する場合は、引き続き、テスト アプリケーションを適切な状態にするステップを記録します。記録を停止し、別の RunScript コマンドを挿入します。
5. 目的のテストをすべて挿入したら、テスト アプリケーションを閉じるためのステップを記録します。他のテストまたはテスト スイートを実行する場合は、それに適した状態までテスト アプリケーションを戻すためのステップを記録します。

手動によるスクリプト作成

記録機能を使用するとスクリプトをすばやく効率的に作成できますが、多くの場合、そのスクリプトを再利用可能にしたり、信頼性や効果を高めたりするには、微調整や拡張が必要です。このことを行うために、任意のスクリプトにコードを手動で追加することができます。Silk Test Workbench はスクリプト言語として VB.NET を使用しているため、VB.NET に関する多少の知識が必要です。Silk Test Workbench は VB.NET スクリプト言語を拡張し、堅牢で強力なスクリプト作成機能を提供します。クラス、メソッド、およびプロパティの詳細と、それぞれのテスト スクリプトでの使用方法については、『*Silk Test Workbench* 言語リファレンス』を参照してください。

また、Silk Test Workbench にはその他の機能もあり、VB.NET のスクリプト機能を利用して手動によるスクリプト作成をより容易にしています。これらの機能には以下のものがあります。

- **オブジェクトの識別** ダイアログ ボックス : **オブジェクトの識別** ダイアログ ボックスでは、ロケータの記録方法や、オブジェクトに使用できるプロパティを表示できます。
- **オートコンプリートおよび構文に関するヒント** : 手動によるコーディングを容易にします。オートコンプリートおよび構文に関するヒントを使用すると、VB.NET 言語要素に自動的にアクセスし、表示して、自動的に入力することができます。これには、すべてのオブジェクトのキーワード、列挙、ブール値、コード構文、使用可能なメソッド、および使用可能な VB.NET プロパティが含まれます。

 **注:** 新しいテスト スクリプトを手動で作成する場合、使用するクラスの拡張をインポートするステートメントをスクリプトに追加する必要があります。たとえば、Win32 メソッドの場合、次の行をスクリプトの開始部分に挿入する必要があります :

```
Imports SilkTest.Ntf.Win32
```


アクセス可能なコントロールでのクリックを記録すると、インポートステートメントがスクリプトに自動的に挿入されます。共通 Silk Test クラス (BaseGuiTestObject など) は常に使用できるため、インポートする必要はありません。

スクリプトの記録

Silk Test Workbench を使用して、アプリケーションをテストするときに実行する操作を記録します。ある操作を実行すると、その操作とそれに対するアプリケーションの応答がコマンドとしてスクリプトに記録されます。

記録中は、記録を停止するまでテストアプリケーションでのすべての操作 (Silk Test Workbench の操作は除く) が Silk Test Workbench によって記録されます。記録が完了したら、生成したスクリプトを変更して、ステップを追加したり、不要なステップを削除したりできます。

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。

 **注:** デフォルトでは、Common プロジェクトにスクリプトが作成されます。別のプロジェクトにスクリプトを作成するには、**プロジェクト** リストで該当するプロジェクトを選択します。

Silk Test Workbench を Oracle データベースで使用する場合は、保存されたすべてのスクリプト名の太文字と小文字が区別されます。つまり、*SCRIPT1* という名前のスクリプトと *Script1* という名前のスクリプトは同じ Oracle データベースに共存でき、重複とはみなされません。ただし、Silk Test Workbench を SQL Server または Access で使用する場合は、保存されたスクリプトの太文字と小文字は区別されません。

3. 省略可能 : Silk Test Workbench の記録オプションを変更する場合は、**オプション** をクリックします。
4. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
5. **OK** をクリックして、スクリプトを資産として保存し、記録を開始します。

記録の開始 チェック ボックスをオンにせずに **OK** をクリックすると、Silk Test Workbench によりスクリプトが資産として保存され、スクリプト テンプレートが表示されます。

アプリケーションの選択 ダイアログ ボックスが開きます。

6. 現在のプロジェクトに対してアプリケーション構成が設定されていない場合、テストするアプリケーションの種類に対応するタブを選択します。
 - ブラウザで実行しない標準アプリケーションをテストする場合は、**Windows** タブを選択します。
 - Web アプリケーションまたはモバイル Web アプリケーションをテストする場合は、**Web** タブを選択します。
 - ネイティブ モバイル アプリケーションをテストする場合は、**モバイル** タブを選択します。
7. 標準アプリケーションをテストするには、現在のプロジェクトに対してアプリケーション構成が設定されていない場合は、リストからアプリケーションを選択します。
8. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。
 - a) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザのインスタンスが既に実行されている場合、**実行中のブラウザーの URL を使用する** をクリックして、実行中のブラウザ インスタンスに現在表示されている URL の記録を行うことができます。
 - b) 省略可能 : あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。

たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザー ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能 : ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能 : **ブラウザー サイズの編集** をクリックすると、新しいブラウザー サイズを指定したり、**ブラウザー サイズ** リストに表示するブラウザー サイズを選択することができます。

9. ネイティブ モバイル アプリケーション (アプリ) をテストするには :

- a) アプリをテストするモバイル デバイスをリストから選択します。
- b) ネイティブ モバイル アプリケーションを選択します。

- モバイル デバイスまたはエミュレータ上にアプリをインストールする場合は、**参照** をクリックしてアプリ ファイルを選択するか、**アプリ ファイル** テキスト フィールドにアプリ ファイルへの完全パスを入力します。このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
- Android デバイス上に既にインストールされているアプリを使用する場合は、**パッケージ/アクティビティ** リストからアプリを選択するか、**パッケージ/アクティビティ** フィールドにパッケージとアクティビティを指定します。
- iOS デバイス上に既にインストールされているアプリを使用する場合は、**バンドル ID** を指定します。
- UFT Mobile 上で利用可能なアプリを使用する場合は、**アプリ識別子** を指定します。

10OK をクリックします。Silk Test Workbench が最小化され、アプリケーションと **記録中** ウィンドウが開きます。

11 テストする操作を記録します。

記録中は、タスクバーの Silk Test Workbench アイコンが点滅します。記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。

12 サポートするブラウザー上で Web アプリケーションを記録している場合で、かつ UI で直接アクセスできないコントロールを扱いたい場合 (たとえば、コントロールが他のコントロールで隠されている場合)、**記録中** ウィンドウの **階層ビューの切り替え** をクリックして、コントロール階層ツリーからコントロールを選択します。

この機能は、次のブラウザーに対するテストで利用できます。

- Microsoft Edge
- Apple Safari
- Mozilla Firefox 41 以降
- Google Chrome 50 以降
- モバイル ブラウザー

13 省略可能 : 検証ロジックをテストに追加します。

- 標準アプリケーションまたは Web アプリケーションをテストしている場合、テストに検証ロジックを追加するには、**Ctrl+Alt** をクリックします。Silk Test Workbench により記録が一時的に停止し、**テスト ロジック デザイナー** ウィザードが表示されます。

ウィザードの指示に従って処理を行い、**完了** をクリックしてウィザードを閉じ、記録を続行します。

- Microsoft Edge 上の Web アプリケーション、またはモバイル アプリケーションをテストしている場合、検証したいオブジェクトをクリックし、**操作の選択** ダイアログ ボックスで **検証の追加** をクリックします。

検証の追加に関する詳細については、「記録中のスクリプトへの検証の追加」を参照してください。

14 Alt+F10 を押すか、**記録中** ウィンドウで **停止** をクリックするか、Silk Test Workbench のタスクバー アイコンをクリックして、記録を停止します。**記録完了** ダイアログ ボックスが開きます。**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、記録が停止したあとにこのダイアログ ボックスは表示されません。この場合、スクリプトが表示されます。

15 次のいずれか 1 つのステップを行います :

- **記録完了** ダイアログ ボックスを閉じて、スクリプトを保存するには、**再生** をクリックします。
再生 ダイアログ ボックスが開き、結果情報を指定してスクリプトを再生できます。
- **コード** ウィンドウでスクリプトを開くには、**.NET スクリプトへ移動** をクリックします。
- スクリプトを保存して、**記録完了** ダイアログ ボックスを終了するには、**保存** をクリックします。
スクリプトが **コード** ウィンドウに表示されます。

記録されたテストがコードウィンドウに表示されているときには、いつでもそのテストを再生できます。

複数のテスト アプリケーションをテストするスクリプトの記録

複数のテスト アプリケーションをテストするスクリプトを記録できます。たとえば、データベースを変更するアプリケーションをテストしているときに、データベースの内容を確認するためにデータベースのビューアー ツールを使用する場合には、そのデータベースのビューアー ツール用のアプリケーション構成を追加する必要があります。

1. テストする主要なアプリケーション用に、スクリプトを記録するか手動で作成します。
2. **プロパティ** ペインで、**アプリケーション構成の追加** を右クリックして選択します。**アプリケーションの選択** ダイアログ ボックスが開きます。
3. 現在のプロジェクトに対してアプリケーション構成が設定されていない場合、テストするアプリケーションの種類に対応するタブを選択します。
 - ブラウザで実行しない標準アプリケーションをテストする場合は、**Windows** タブを選択します。
 - Web アプリケーションまたはモバイル Web アプリケーションをテストする場合は、**Web** タブを選択します。
 - ネイティブ モバイル アプリケーションをテストする場合は、**モバイル** タブを選択します。
4. 標準アプリケーションをテストするには、現在のプロジェクトに対してアプリケーション構成が設定されていない場合は、リストからアプリケーションを選択します。
5. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。
 - a) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。選択したブラウザのインスタンスが既に実行されている場合、**実行中のブラウザの URL を使用する** をクリックして、実行中のブラウザ インスタンスに現在表示されている URL の記録を行うことができます。
 - b) 省略可能：あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザ ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
6. ネイティブ モバイル アプリケーション (アプリ) をテストするには：
 - a) アプリをテストするモバイル デバイスをリストから選択します。
 - b) ネイティブ モバイル アプリケーションを選択します。
 - モバイル デバイスまたはエミュレータ上にアプリをインストールする場合は、**参照** をクリックしてアプリ ファイルを選択するか、**アプリ ファイル** テキスト フィールドにアプリ ファイルへの完全パスを入力します。このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
 - Android デバイス上に既にインストールされているアプリを使用する場合は、**パッケージ/アクティビティ** リストからアプリを選択するか、**パッケージ/アクティビティ** フィールドにパッケージとアクティビティを指定します。
 - iOS デバイス上に既にインストールされているアプリを使用する場合は、**バンドル ID** を指定します。
 - UFT Mobile 上で利用可能なアプリを使用する場合は、**アプリ識別子** を指定します。
7. **OK** をクリックします。
8. 新しいアプリケーション構成を使用して、スクリプトの追加ステップを記録します。

Java Network Launching Protocol (JNLP) を使用するアプリケーションを起動するスクリプトの構成

Java Network Launching Protocol (JNLP) を使用して起動するアプリケーションでは、Silk Test Workbench に追加の構成が必要です。これらのアプリケーションは Web から起動されるため、実際のアプリケーションと、「Web Start」を起動するアプリケーションを起動するように、アプリケーション構成を手動で構成する必要があります。このようにしない場合、アプリケーションがすでに実行されていないかぎり、ビジュアル テストを再生すると、失敗します。

1. Silk Test Workbench がアプリケーションを起動できずにテストが失敗する場合は、アプリケーション構成を編集します。
2. アプリケーションを起動するコードを含めるスクリプトを開きます。
3. プロパティ ペインで、**アプリケーション構成** カテゴリに移動します。
4. 変更するアプリケーション構成をダブルクリックします。 **アプリケーション構成の編集** ダイアログ ボックスが表示されます。
5. 基本状態を編集して、再生中に Web Start が起動されるようにします。
 - a) **編集** をクリックします。
 - b) **実行可能ファイル パターン** テキスト ボックスに、javaws.exe への絶対パスを入力します。
たとえば、以下のように入力します。

```
%ProgramFiles%\Java\jre6\bin\javaws.exe
```
 - c) **コマンドラインパターン** テキスト ボックスに、Web Start への URL を含むコマンドラインパターンを入力します。

```
"<url-to-jnlp-file>"
```


たとえば、SwingSet3 アプリケーションの場合、以下のように入力します。

```
"http://download.java.net/javadesktop/swingset3/SwingSet3.jnlp"
```
 - d) **OK** をクリックします。
6. **OK** をクリックします。テストは、基本状態を使用し、Web 起動アプリケーションとアプリケーション構成の実行可能パターンを開始して javaw.exe に接続し、テストを実行します。

テストを実行すると、アプリケーション構成の EXE ファイルが基本状態の EXE ファイルと一致しないという警告が表示されます。テストは予想したとおりに実行されているため、このメッセージは無視できません。

新しいブラウザ ウィンドウで開くスクリプトの構成

スクリプト用に既存のブラウザ ウィンドウを使用するのではなく、別のブラウザ ウィンドウを開く場合があります。たとえば、Silk Central Test Manager から Silk Test Workbench スクリプトを実行する場合には、スクリプトは既存の Silk Central ウィンドウを使用します。Silk Central に戻るには、ユーザーはブラウザ ウィンドウで **戻る** をクリックする必要があります。Silk Central から実行するテスト用に別のブラウザ ウィンドウを開くスクリプトを構成すると、この動作を回避できます。

1. アプリケーションを起動するコードを含めるスクリプトを開きます。
2. スクリプトで基本状態を使用する場合は、以下のステップを実行します。
 - a) プロパティ ペインで、**アプリケーション構成** カテゴリに移動します。
 - b) 変更するアプリケーション構成をダブルクリックします。 **アプリケーション構成の編集** ダイアログ ボックスが開きます。
 - c) **基本状態の実行** チェック ボックスをオフにします。
3. スクリプトの先頭に Imports System.Diagnostics と入力します。
4. アプリケーションを新しいプロセスとして定義するには、以下のように入力します。

```
Dim applicationProcess As New Process()
```

`applicationProcess` は、起動するアプリケーションの名前です。

たとえば、Internet Explorer を起動するには、以下のように入力します。

```
Dim ieProcess As New Process()
```

5. 以下のコードを入力して、アプリケーションのフルパスを定義します。

```
Dim psi As New ProcessStartInfo()  
psi.FileName = "applicationexecutable.exe"
```

`applicationexecutable.exe` は、テスト アプリケーションを起動する実行可能ファイルの名前です。たとえば、Internet Explorer を起動するには、以下のように入力します。

```
Dim psi As New ProcessStartInfo()  
psi.FileName = "C:\Program Files\Internet Explorer\iexplore.exe"
```



注: テストするアプリケーションの実行可能ファイルが Silk Test Workbench と同じディレクトリにある場合は、フルパスを指定する必要はありません。

6. テストする Web アプリケーションを開くには、`ProcessStartInfo.Arguments` プロパティを URL に設定します。

たとえば、Internet Explorer を起動してサンプル Web アプリケーションをテストするには、以下の行を `Filename` 行のあとに追加します。

```
psi.Arguments = "http://demo.borland.com/InsuranceWebExtJS/index.jsf"
```

7. プロセスを起動するコードを指定します。

たとえば、`ieProcess` を起動するには、以下のように入力します。

```
ieProcess.StartInfo = psi  
ieProcess.Start()
```

8. スクリプトを保存します。

テスト アプリケーションでのコントロールの識別 (.NET スクリプト)

.NET スクリプトを記録する場合、Silk Test Workbench によって自動的にコントロールが識別されます。既存のコントロールを変更したり、手動で操作を挿入したりするには、テストするアプリケーションでコントロール名を識別します。

1. ツールバーの **オブジェクトの識別** をクリックします。**オブジェクトの識別** ダイアログ ボックスが開きます。
2. **識別の開始** をクリックします。
3. テストしているアプリケーションにカーソルを移動します。カーソルをアプリケーション内の各コントロール上に動かすたびにコントロールが強調表示されます。関連するロケータ文字列またはオブジェクト マップ項目は、**選択済みロケータ** テキスト ボックスに表示されます。
4. 使用するコントロールが強調表示されているときに、コントロールをクリックします。
5. 省略可能：**ロケータの詳細の表示** をクリックすると、ロケータ ツリーや関係する属性のすべてが **ロケータ属性** テーブルに表示されます。
6. 省略可能：記録したロケータ属性は、**ロケータ属性** テーブルの別のロケータ属性で置き換えることができます。

たとえば、記録したロケータは以下のように表示されます。

```
/BrowserApplication//BrowserWindow//input[@id='loginButton']
```

ロケータ属性 テーブルに `textContent Login` がリストされている場合、以下のようにしてロケータを手動で変更できます。

```
/BrowserApplication//BrowserWindow//input[@textContent='Login']
```

新しいロケータは、**選択済みロケータ** テキスト ボックスに表示されます。

7. **テスト** をクリックして、ロケーターがコントロールを正しく認識することを確認します。
8. ロケーターをスクリプトにコピーするには、**貼り付け** をクリックします。
9. **閉じる** をクリックします。

スクリプト内からのアプリケーションの起動

通常、スクリプトを記録すると、テストするアプリケーションを起動する基本状態が自動的に記録されます。ただし、スクリプトで複数のアプリケーションをテストするか、何らかの理由で **基本状態の実行** 機能をオフにする場合は、スクリプト内からテスト アプリケーションを起動する必要があります。

1. アプリケーションを起動するコードを含めるスクリプトを開きます。
2. スクリプトの先頭に `Imports System.Diagnostics` と入力します。
3. アプリケーションを新しいプロセスとして定義するには、以下のように入力します。

```
Dim applicationProcess As New Process()
```

`applicationProcess` は、起動するアプリケーションの名前です。
たとえば、Internet Explorer を起動するには、以下のように入力します。


```
Dim ieProcess As New Process()
```

4. 以下のコードを入力して、アプリケーションのフルパスを定義します。

```
Dim psi As New ProcessStartInfo()  
psi.FileName = "applicationexecutable.exe"
```

`applicationexecutable.exe` は、テスト アプリケーションを起動する実行可能ファイルの名前です。たとえば、Internet Explorer を起動するには、以下のように入力します。

```
Dim psi As New ProcessStartInfo()  
psi.FileName = "C:¥Program Files¥Internet Explorer¥iexplore.exe"
```

 **注:** テストするアプリケーションの実行可能ファイルが Silk Test Workbench と同じディレクトリにある場合は、フルパスを指定する必要はありません。

5. 初期化時に特定の Web ページに移動するには、URL に `ProcessStartInfo.Arguments` プロパティを設定します。
たとえば、Internet Explorer を起動して Google.com に移動するには、Filename 行のあとに以下の行を追加します。

```
psi.Arguments = "http://www.google.com"
```

6. プロセスを起動するコードを指定します。
たとえば、ステップ 1 で定義した `ieProcess` を起動するには、以下のように入力します。

```
ieProcess.StartInfo = psi  
ieProcess.Start()
```

7. スクリプトを保存します。

Internet Explorer を起動して Google.com に移動するスクリプト全体は、以下のようになります。

```
Imports System  
Imports System.Diagnostics
```

```
Public Module Main
```

```
Public Sub Main()
```

```
Dim _desktop As Desktop = Agent.Desktop  
Dim ieProcess As New Process()  
Dim psi As New ProcessStartInfo()  
psi.FileName = "C:¥Program Files¥Internet Explorer¥iexplore.exe"  
psi.Arguments = "http://www.google.com"
```

```
ieProcess.StartInfo = psi
ieProcess.Start()

End Sub
End Module
```

スクリプトのオブジェクト マップ項目またはロケータの 手動による記録

開始する前に、テストするアプリケーションが実行中であることを確認します。

オブジェクトの識別 ダイアログ ボックスを使用して、オブジェクト マップ項目またはロケータを手動でキャプチャします。**オブジェクトの識別** ダイアログ ボックスを使用すると、テストするオブジェクトを簡単かつ確実に識別することができます。

1. オブジェクトを含めるスクリプトを開き、オブジェクトの挿入先となるスクリプト内の場所をクリックします。
2. ツール > **オブジェクトの識別** を選択します。**オブジェクトの識別** ダイアログ ボックスが開きます。
3. **選択モード** を指定します。

- **クリック** : オブジェクトをクリックして、ロケータを識別します。
- **ホット キー** : **キー入力** リスト ボックスで指定されたキー入力の組み合わせを使用してオブジェクトをキャプチャするには、このモードを指定します。通常、このモードは、ユーザーがクリックしたときにだけ表示されるメニューやコンボ ボックスなどのオブジェクトをキャプチャする場合に選択します。このモードでは、オブジェクトを選択し、ホット キー入力の組み合わせを押すことによって、オブジェクトを閉じることなくロケータをキャプチャします。

4. **識別の開始** をクリックします。

Google Chrome でオブジェクト マップ項目またはロケータをキャプチャする場合は、Google Chrome の選択されたインスタンスが適切なオートメーション パラメータを使用して起動されているかどうかを Silk Test Workbench が検出します。適切なパラメータを使用して起動されていない場合は、Silk Test Workbench が Google Chrome を終了し、オートメーション パラメータ セットを使用して再起動します。

5. 省略可能 : ロケータを記録する前に、テスト対象アプリケーションを特定の状態にするには、**識別の停止** をクリックします。これ以降、テスト対象アプリケーションで実行する操作は、記録されなくなります。ロケータの記録を再開する場合は、**識別の開始** をクリックします。
6. 記録したいオブジェクト上にマウスを移動して、次のステップのいずれかを行います :

- **クリック** モードを使用する場合は、識別するオブジェクトをクリックします。
- キー入力の組み合わせを押して、**ホット キー** モードでオブジェクトをキャプチャします。
デフォルトでは、キー入力の組み合わせは Ctrl+Shift です。

Silk Test Workbench によって、関連するロケータ文字列が **選択済みロケータ** テキスト ボックスに表示されます。

7. **ロケータの詳細** テーブルでのロケータの表示方法を絞り込むには、以下のいずれかの操作を実行します。

- **ロケータの詳細の非表示** : **ロケータの詳細** テーブルを非表示にするには、このリンクをクリックします。
- **オブジェクト マップの表示** : オブジェクト マップ項目名を **ロケータ** 列に表示するには、このチェック ボックスをオンにします。オブジェクト マップ項目名は、コントロールまたはウィンドウに対して、コントロールやウィンドウのロケータではなく論理名 (エイリアス) を関連付けます。デフォルトでは、オブジェクト マップ項目名が表示されます。ロケータを使用するには、このチェック ボックスをオフにします。

- **完全ロケータの表示** : 完全ロケータ名を表示するには、このチェックボックスをオンにします。オブジェクトに関連付けられた属性のみを表示するには、このチェックボックスをオフにします。
 - **プロパティの表示** : **ロケータの詳細** テーブルで選択したオブジェクトの属性および属性値を表示するには、このチェックボックスをオンにします。このテーブルの属性を選択して、ロケータの識別に使用できます。[プロパティ] サブツリーを非表示にし、ロケータの詳細のみを表示するには、このチェックボックスをオフにします。
8. **選択済みロケータ** テキストボックスに表示されるオブジェクトが使用するオブジェクトであることをテストするには、**テスト** をクリックします。Silk Test Workbench により、ロケータに対応するオブジェクトが、テストするアプリケーションで強調表示されます。
 9. 記録したロケータを置換するには、**ロケータの詳細** テーブルで、使用するロケータを選択します。新しいロケータは、**選択済みロケータ** テキストボックスに表示されます。
 10. ロケータをスクリプトにコピーするには、**貼り付け** をクリックします。Silk Test Workbench によって、ロケータがスクリプトに追加されます。
 11. スクリプトで正しくコンパイルされ、実行する操作が含まれるように、ロケータを変更します。たとえば、以下の完全ロケータをスクリプトに貼り付けます。

```
/WPFWindow[@caption='Basic Controls'][1]//
WPFRichTextBox[@automationId='editableTextBox']
```

With ステートメントに /WPFWindow 宣言がすでに含まれている場合は、以下のロケータを使用し、そのあとにメソッドが続くようにスクリプトを変更します。

```
.WPFRichTextBox("@automationId='editableTextBox']").Select(1,10)
```

既存のテストへの追加操作の記録

テストを作成したあと、テストを開き、テストの任意の場所から追加操作を記録できます。これにより、既存のテストを追加操作で更新できます。

1. 既存のテスト スクリプトを開きます。
2. 追加操作を記録するテスト スクリプトの場所を選択します。



注: 記録した操作は、選択した場所の後に挿入されます。テスト対象アプリケーション (AUT) は基本状態に戻りません。代わりに、テスト スクリプトの直前の操作が記録された範囲で AUT を開いておきます。

3. **操作 > 記録** をクリックします。

Silk Test Workbench が最小化され、**記録中** ダイアログ ボックスが開きます。

4. AUT に対して実行したい追加操作を記録します。

記録中は、タスクバーの Silk Test Workbench アイコンが点滅します。記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。

5. 記録を停止するには、**記録中** ウィンドウの **停止** をクリックします。

記録を停止するキーの組み合わせ (デフォルトでは **Alt+F10**) を押すか、タスクバーの Silk Test Workbench アイコンをクリックして、記録を停止することもできます。

既存のスクリプトを開く

既存のスクリプトを開いて、再生、確認、または変更を行います。

1. **ファイル > 開く** を選択します。 **アセット ブラウザ** が開きます。
2. 左側のペインで **.NET スクリプト** を選択し、スクリプトのリストを表示します。
3. 右側のペインで、開く資産をダブルクリックします。

スクリプトの編集

スクリプトを作成して保存したあとは、いつでもそれを編集できます。

1. スクリプトを開きます。
2. テキスト エディタの場合と同様にスクリプトを編集します。
テキストの選択、削除、切り取り、コピー、貼り付けなどが可能です。スクリプト ウィンドウには、オートコンプリートと構文の関連付けテクノロジーも備えられており、少ないエラーでコードを完成するのに役立ちます。
たとえば、*Window* のあとにピリオドを入力すると、Window オブジェクトに有効なメソッドとプロパティのリストがすべて表示されます。
同様に、有効な関数のあとにスペース、カンマ、または左かっこを入力すると、構文に関するヒントが表示されます。
3. ツールバーの **保存** をクリックするか、または **ファイル > 保存** を選択して、説明を入力し、スクリプトを保存します。

スクリプトの保存

新規スクリプトまたは変更したスクリプトを保存できます。更新したスクリプトを閉じると、変更内容を保存するかどうかを確認するプロンプトが表示されます。

1. **ファイル > 名前を付けて保存** を選択します。 **名前を付けて保存** ダイアログ ボックスが開きます。
2. 省略可能：スクリプト名を変更し、説明を入力します。
3. **OK** をクリックします。

スクリプトでの要素の定義への移動

VB .NET スクリプトでメソッドやクラスの定義に移動するには、次のアクションを実行します。

1. スクリプトを開きます。
2. 定義に移動する要素を右クリックします。
代わりに、**F12** を押すこともできます。
3. **定義へ移動** を選択します。


アプリケーション構成

アプリケーション構成は、テストするアプリケーションに Silk Test Workbench が接続する方法を定義します。Silk Test Workbench は、基本状態を作成するときに、アプリケーション構成を自動的に作成します。しかし、アプリケーション構成を追加したり、変更や削除をすることが必要になる場合があります。たとえば、データベースを変更するアプリケーションをテストしているときに、データベースの内容を確認するためにデータベースのビューアー ツールを使用する場合には、そのデータベースのビューアー ツール用のアプリケーション構成を追加する必要があります。


- Windows アプリケーションの場合、アプリケーション構成には以下が含まれます。
 - 実行可能ファイル パターン
このパターンに一致するすべてのプロセスは、テストに対して有効化されます。たとえば、Internet Explorer の実行可能パターンは *¥IEXPLORE.EXE です。実行可能ファイルの名前が IEXPLORE.EXE で、任意のディレクトリに置かれているプロセスはすべて有効化されます。
 - コマンドライン パターン
コマンドライン パターンは、テストを行うために有効化されるプロセスの制約に使用される補足パターンで、コマンドライン引数の一部 (実行可能ファイル名の後ろ部分) をマッピングすることにより行います。コマンドライン パターンを含むアプリケーション構成では、実行可能パターンとコマンドライン パターンの両方に一致するプロセスのみが、テストに対して有効化されます。コマンドライン パターンが定義されていない場合は、指定された実行可能ファイル パターンを持つすべてのプロセスが有効化されます。コマンドラインの使用は、Java アプリケーションに対して特に有益で

す。これは、ほとんどの Java プログラムが javaw.exe を使用して実行されるためです。つまり、典型的な Java アプリケーションに対してアプリケーション構成を作成する場合、実行可能パターンには *¥javaw.exe が使用され、このパターンはすべての Java プロセスに一致します。このような場合、コマンドラインパターンを使用して、該当するアプリケーションのみがテストに対して有効化されるようにします。たとえば、アプリケーションのコマンドラインが **com.example.MyMainClass** で終わる場合には、コマンドラインパターンに ***com.example.MyMainClass** を使用します。

- ローカルマシン上のデスクトップブラウザの Web アプリケーションの場合、アプリケーション構成にはブラウザの種類だけが含まれます。

 **注:** コマンドライン引数を指定してブラウザを開始したり、ブラウザの作業ディレクトリや実行可能ファイルを指定するには、ブラウザの種類でカスタムを選択します。詳細については、「基本状態の変更」を参照してください。

- リモートマシン上の Apple Safari や Microsoft Edge の Web アプリケーションの場合、アプリケーション構成には以下が含まれます。
 - ブラウザの種類
 - 接続文字列
- モバイルブラウザの Web アプリケーションの場合、アプリケーション構成には以下が含まれます。
 - ブラウザの種類
 - 接続文字列
- ネイティブモバイルアプリケーションの場合、アプリケーション構成には以下が含まれます。
 - 接続文字列
 - シンプルなアプリケーション名。モバイルデバイス上で複数のアプリケーションが同じ名前を持つ場合は、アプリケーションの完全修飾名が使用されます。

 **注:** 定義済みの基本状態を使って Web アプリケーションをテストするときに、ブラウザーアプリケーション構成を複数追加しないでください。

アプリケーション構成の変更

アプリケーション構成は、テストするアプリケーションに Silk Test Workbench が接続する方法を定義します。Silk Test Workbench は、基本状態を作成するときに、アプリケーション構成を自動的に作成します。しかし、アプリケーション構成を追加したり、変更や削除をすることが必要になる場合があります。たとえば、データベースを変更するアプリケーションをテストしているときに、データベースの内容を確認するためにデータベースのビューアー ツールを使用する場合には、そのデータベースのビューアー ツール用のアプリケーション構成を追加する必要があります。

Silk Test Workbench で使用するアプリケーション構成が、プロジェクト全体に関連付けられるか、特定のビジュアル テストまたは VB .NET スクリプトに関連付けられるかは、テストタイプによって異なります。

- プロジェクト全体に関連付けられているアプリケーション構成を変更するには：
 - Silk Test Workbench メニューから **ツールアプリケーション構成の編集** を選択します。 **プロジェクトの選択** ダイアログが開きます。
 - リストからプロジェクトを選択します。
 - 編集** をクリックします。 **アプリケーション構成の編集** ダイアログが開きます。
 - 該当するアプリケーション構成の隣にある **編集** をクリックします。
- 特定のビジュアル テストに関連付けられているアプリケーション構成を変更するには：
 - ビジュアル テストを開きます。
 - タスク** ペインで **<<開始>>** ステップをクリックします。
 - プロパティ** ペインの **アプリケーション構成** カテゴリに移動します。
 - 変更するアプリケーション構成をクリックします。

- e) **アプリケーション構成の編集** をクリックします。
3. 特定の VB .NET スクリプトに関連付けられているアプリケーション構成を変更するには：
- a) VB .NET スクリプトを開きます。
 - b) **プロパティ** ペインの **アプリケーション構成** カテゴリに移動します。
 - c) 変更するアプリケーション構成をダブルクリックします。
4. プロジェクト全体に関連付けるアプリケーション構成をさらに追加するには、**アプリケーション構成の追加** をクリックします。



注: 定義済みの基本状態を使って Web アプリケーションをテストするときに、ブラウザ アプリケーション構成を複数追加しないでください。

アプリケーションの選択 ダイアログ ボックスが開きます。タブを選択してからテストするアプリケーションを選択して **OK** をクリックします。

5. 特定のビジュアル テストに関連付けるアプリケーション構成をさらに追加するには：
- a) ビジュアル テストを開きます。
 - b) **タスク** ペインで **<<開始>>** ステップをクリックします。
 - c) **プロパティ** ペインの **アプリケーション構成** カテゴリに移動します。
 - d) **アプリケーション構成の追加** をクリックします。
6. 特定の VB .NET スクリプトに関連付けるアプリケーション構成をさらに追加するには：
- a) VB .NET スクリプトを開きます。
 - b) **プロパティ** ペインで右クリックします。
 - c) **アプリケーション構成の追加** を選択します。
7. プロジェクト全体に関連付けられているアプリケーション構成を削除するには、該当するアプリケーション構成の隣にある **削除** をクリックします。
8. 特定のビジュアル テストに関連付けられているアプリケーション構成を削除するには：
- a) ビジュアル テストを開きます。
 - b) **タスク** ペインで **<<開始>>** ステップをクリックします。
 - c) **プロパティ** ペインの **アプリケーション構成** カテゴリに移動します。
 - d) 削除するアプリケーション構成をクリックします。
 - e) **アプリケーション構成の削除** をクリックします。
9. 実行可能ファイルを指定するには、**実行可能ファイル** フィールドに実行可能ファイルへの完全パスを入力します。



注: Web アプリケーションをテストする場合に、ブラウザの実行可能ファイルを指定する場合は、ブラウザの種類からカスタムを選択します。

たとえば、Mozilla Firefox を起動するには、「C:\Program Files (x86)\Mozilla Firefox\firefox.exe」を入力します。

- 10 デスクトップ アプリケーションのテストを行う場合は、**実行可能ファイル パターン** フィールドに、テストするアプリケーションの実行可能ファイルの名前を入力します。
- パターンでは大文字と小文字が区別されず、任意の長さの任意のテキストに一致するワイルドカードとしてアスタリスク (*) を、1 文字に一致するワイルドカードとして疑問符 (?) を使用できます。
- たとえば、メモ帳 (Notepad) に対する実行可能ファイル パターンは、*%notepad.exe になります。実行可能ファイルの名前が notepad.exe で、任意のディレクトリに置かれているプロセスはすべて有効化されます。
- 11 デスクトップ アプリケーションをテストし、実行可能ファイルと一緒にコマンド ライン パターンを使用したい場合には、コマンド ライン パターンを **コマンド ライン パターン** テキスト ボックスに入力します。
- 12 Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。
- 変更** をクリックして **アプリケーションの選択** ダイアログ ボックスを開き、使用するブラウザを選択することもできます。

- 13 Web アプリケーションをテストする場合は、**移動する URL** テキストボックスに、Web アプリケーションのアドレスを入力します。
- 14 Web アプリケーションのビジュアル ブレークポイントを検出するには、**ブラウザー サイズの編集** をクリックします。
詳細については、「ビジュアル ブレークポイントの検出」を参照してください。
- 15 省略可能：あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザー ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
- 16 省略可能：ブラウザー ウィンドウの **向き** を選択します。
- 17 省略可能：**ブラウザー サイズの編集** をクリックすると、新しいブラウザー サイズを指定したり、**ブラウザー サイズ** リストに表示するブラウザー サイズを選択することができます。
- 18 モバイル アプリケーションや Apple Safari 上の Web アプリケーションをテストする場合は、**接続文字列** テキストボックスに接続文字列を入力します。
詳細については、「**モバイル デバイスの接続文字列**」を参照してください。
- 19 **OK** をクリックします。

[アプリケーションの選択] ダイアログ ボックス

アプリケーションの選択 ダイアログ ボックスを使用して、テストしたアプリケーションを選択し、アプリケーションとオブジェクト マップを関連付けたり、アプリケーション構成をテストに追加したりします。アプリケーションの種類は、ダイアログ ボックスのタブとしてリストされます。使用したいアプリケーションの種類に対応したタブを選択します。

Windows システムで実行中のすべての Microsoft Windows アプリケーションの一覧が表示されます。リストから項目を選択して、**OK** をクリックします。

キャプションを持たないプロセスを表示しない チェック ボックスを使用して、キャプションを持たないアプリケーションを一覧から除去します。

Web

利用可能なすべてのブラウザーの一覧が表示されます (任意の接続済みモバイル デバイス上のモバイル ブラウザーや Mac 上の Apple Safari を含む)。**移動する URL の入力** テキストボックスに、開く Web ページを指定します。選択したブラウザーのインスタンスが既に実行されている場合、**実行中のブラウザーの URL を使用する** をクリックして、実行中のブラウザー インスタンスに現在表示されている URL の記録を行うことができます。あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。ブラウザー サイズを選択する場合は、選択したブラウザーの **向き** も選択できます。



注: 定義済みの基本状態を使って Web アプリケーションをテストするときに、ブラウザー アプリケーション構成を複数追加しないでください。

モバイル

利用可能なすべてのモバイル デバイスとすべての実行中の Android エミュレーターの一覧が表示されます (リモート ロケーションに接続されたデバイスを含む)。モバイル ネイティブ アプリケーションをテストする場合は、このタブを選択します。選択したモバイル デバイス上で現在実行中のモバイル アプリケーション (アプリ) をテストするために選択したり、テストするアプリの名前やファイルを手動で参照したり指定することができます。

リモート ロケーションの編集

リモート ロケーション ダイアログ ボックスを使用すると、リモート ロケーション上のブラウザーやモバイル デバイスを、テストするアプリケーションのセットに追加できます。

1. ツール > **リモート ロケーションの編集** をクリックします。 **リモート ロケーション** ダイアログ ボックスが表示されます。

2. リモート ロケーションを追加するには、次の操作を実行します。

- a) **ロケーションの追加** の右側にある矢印をクリックして、Silk Test Information Service、Silk Central、UFT Mobile のいずれかを選択して、追加するリモート ロケーションを指定します。



注: Silk Central は 1 つだけリモート ロケーションとして設定できます。Silk Central との統合が既に設定されている場合は、リモート ロケーション リストに Silk Central が表示されません。

- b) **ロケーションの追加** をクリックします。**ロケーションの追加** ダイアログ ボックスが表示されます。

- c) Silk Test Workbench がリモート マシン上の Information Service に接続するためのリモート ロケーションの URL とポートを、**ホスト** フィールドに入力します。

デフォルトのポートは 22901 です。

- d) UFT Mobile との接続を追加する際に、HTTPS によるセキュアな接続を使用する場合は、UFT Mobile 証明書をインストールします。

詳細については、「[UFT Mobile との HTTPS 接続用証明書のインストール](#)」を参照してください。

- e) UFT Mobile との接続を追加する場合は、UFT Mobile の **ユーザー名** と **パスワード** を指定します。

UFT Mobile パスワードの変更については、「[UFT Mobile パスワードの変更](#)」を参照してください。

- f) 省略可能：**名前** フィールドにリモート ロケーションの名前を入力します。

- g) 省略可能：**テスト** をクリックして、リモート接続が正しく動作することを確認します。

- h) **OK** をクリックします。

3. 既存のリモート ロケーションを編集するには、**編集** をクリックします。

4. リモート ロケーションを削除するには、**削除** をクリックします。

5. 省略可能：**アプリケーションの選択** ダイアログに表示されるブラウザーやデバイスの数を減らすには、**このロケーション上のデバイスとブラウザーを非表示** をクリックします。そのリモート ロケーションにインストールされたブラウザーや接続されたデバイスが、**アプリケーションの選択** ダイアログに表示されなくなります。デフォルトでは、すべてのリモート ロケーションにインストールされたブラウザーや接続されたデバイスのすべてが、**アプリケーションの選択** ダイアログに表示されます。

6. **OK** をクリックします。

リモート ロケーションを追加すると、リモート ロケーション上にインストールされているブラウザ (Mac 上の Apple Safari など) が、**アプリケーションの選択** ダイアログ ボックスの **Web** タブで利用可能になり、リモート ロケーションに接続されているモバイル デバイスが、**アプリケーションの選択** ダイアログ ボックスの **モバイル** タブで利用可能になります。

ユーザー アカウント制御

ユーザー アカウント制御 (UAC) を設定して Silk Test Workbench を使用できます。アプリケーションのセキュリティレベルが Silk Test Workbench と同じである場合、UAC が有効な状態でアプリケーションをテストできます。


システムに Silk Test Workbench をインストールするには管理者権限が必要です。Silk Test Workbench がインストールされている場合は、Silk Test Workbench を使用するための管理者権限は必要ありません。



注: UAC が有効な状態で、Silk Test Workbench を使用する場合は、Silk Test Workbench にインストール ディレクトリへの書き込み権限が必要となるため、Silk Test Workbench の Program Files ディレクトリへのインストールは推奨しません。Silk Test Workbench を Program Files ディレクトリにインストールした場合は、Silk Test Workbench のデータベースを構成するために管理者権限が必要です。

特定のテクノロジー ドメインの無効化

あるテストにおいて、特定のテクノロジー ドメインが不要で、問題を起こすことがわかっている場合 (テストのパフォーマンスを低下させる場合など)、そのテクノロジー ドメインをロードしないように無効化できます。

 **注:** そのテクノロジー ドメインがテストに必要なことが確実な場合にのみ、テクノロジー ドメインを無効化してください。

1. ビジュアル テストの **テクノロジー ドメイン** ダイアログ ボックスを開くには：
 - a) ビジュアル テストの <<開始>> ステップを選択します。
 - b) **プロパティ** タブを開きます。
 - c) **テクノロジー ドメイン** フィールドをクリックします。
 - d) **開く** をクリックします。
2. VB .NET スクリプトの **テクノロジー ドメイン** ダイアログ ボックスを開くには：
 - a) **プロパティ** タブを開きます。
 - b) プロパティ ツリーで、テクノロジー ドメインを無効化するアプリケーション構成を右クリックします。
 - c) **テクノロジー ドメインの選択** をクリックします。
3. **テクノロジー ドメイン** ダイアログ ボックスで、**テクノロジー ドメインを自動的にロードする** チェック ボックスをオフにします。
4. **テクノロジー ドメイン** リストから、そのテストでロードする必要の無いテクノロジー ドメインをオフにします。
5. **OK** をクリックします。

テストの以降の実行では、有効なテクノロジー ドメインだけがロードされます。

アプリケーション構成エラー

アプリケーションにアタッチできない場合、以下のエラー メッセージが表示されます。アプリケーション <アプリケーション名> にアタッチするのに失敗しました。詳細については、ヘルプを参照してください。

この場合、以下の表に示されている 1 つ以上の問題が原因である可能性があります。

問題	原因	解決策
タイムアウト	<ul style="list-style-type: none"> • システムが遅すぎます。 • システムのメモリ サイズが小さすぎます。 	速いシステムを使用するか、現在使用しているシステムのメモリ使用量を減らします。
ユーザー アカウント制御 (UAC) の失敗	システムの管理者権限がありません。	管理者権限を持つユーザー アカウントでログインします。
コマンド ライン パターン	コマンド ライン パターンが固有すぎます。この問題は特に Java の場合に発生します。再生が意図したとおりに機能しないことがあります。	パターンから不明瞭なコマンドを削除します。
<ul style="list-style-type: none"> • ブラウザーの選択 ダイアログ ボックスが、Web アプリケーションに対してテストを実行しているときに表示されない • Web アプリケーションに対してテストを実行しているときに、複数のブラウザー インスタンスが開始される • 開いているブラウザー インスタンスを使用して Web アプリケーションに対してテストを実行して 	基本状態と複数のブラウザー アプリケーション構成がテストケースに対して定義されています。	1 つを除くすべてのブラウザー アプリケーション構成をテストケースから削除します。

問題	原因	解決策
<p>いるときに、Silk Test Workbench が動作を停止することがある</p> <p>テスト実行時の再生エラー</p>	<p>テストに対してアプリケーション構成が定義されていません。</p> <p>次の例外が表示される可能性があります：</p> <p>アプリケーション構成が存在しません。</p>	<ul style="list-style-type: none"> キーワード駆動テストを実行するときに、基本状態を実行するキーワードがテストに含まれていることを確認します。 アプリケーション構成が現在のプロジェクトに対して設定されていることを確認します。

アプリケーション構成のトラブルシューティング

アプリケーションの選択 ダイアログ ボックスにアプリケーションが表示されない理由

- **キャプションを持たないプロセスを表示しない** チェック ボックスをオフにします。このチェック ボックスは、デフォルトではオンになっており、キャプションを持たないアプリケーションはダイアログ ボックスに表示されません。
- Silk Test Workbench をシステム特権で実行します。
 1. Silk Test Workbench を閉じます。
 2. Open Agent を停止します。
 3. Silk Test Workbench を管理者として実行します。
- **タスク マネージャー** を使用して、アプリケーションが他のユーザー アカウントで実行されていないか確認します。
- アプリケーションが runas コマンド、または同等のコマンドで起動されていないか確認します。

アプリケーション構成での環境変数

以下の環境変数は、アプリケーション構成で使用できます。

環境変数	説明
TMP または TEMP	Windows 一時ディレクトリ。
USERPROFILE	ユーザー プロファイルディレクトリ (C:¥Documents and Settings¥myUserName)。
APPDATA	アプリケーション データ ディレクトリ。
ProgramFiles	Program Files ディレクトリ。
ProgramFiles(x86)	32 ビット アプリケーションの Program Files ディレクトリ (64 ビット環境でのみ使用可能)。

基本状態

アプリケーションの基本状態とは、各テストの実行開始前にアプリケーションに想定される既知の安定した状態です。大抵の場合この状態は、アプリケーションを最初に起動したときの状態になります。基本状態は、記録するアプリケーションを選択すると自動的に生成されます。

ビジュアル テストまたはスクリプトを記録すると、Silk Test Workbench により基本状態が自動的に作成されます。テストの必要性に応じて基本状態を有効にしたり無効にしたりすることができます。たとえ

ば、アプリケーション ウィンドウを手動で開く場合があります。このような場合には、基本状態を無効にします。基本状態の情報は、アプリケーション構成の一部です。

基本状態はテストの整合性を保障するための重要な一因です。各テストが安定した基本状態から開始できることを保障することによって、あるテストのエラーによって、後続のテストが失敗しないことを保障することができます。

Silk Test Workbench は、ビジュアル テストまたはスクリプトを実行する前に、アプリケーションがその基本状態にあることを保障します。

エラーが発生すると、Silk Test Workbench はテストの実行を停止します。

基本状態のオン/オフの切り替え

ビジュアル テストまたはスクリプトを記録すると、Silk Test Workbench により基本状態が自動的に作成されます。テストの必要性に応じて基本状態を有効にしたり無効にしたりすることができます。たとえば、アプリケーション ウィンドウを手動で開く場合があります。このような場合には、基本状態を無効にします。

1. ビジュアル テストで、以下のステップを実行します。

- a) **テスト ステップ** ペインで、**<<開始>>** ステップをクリックします。
- b) **プロパティ** ペインで、**アプリケーション構成** カテゴリに移動します。
- c) アプリケーション構成フィールドをクリックし、**アプリケーション構成の編集** をクリックします。**アプリケーション構成の編集** ダイアログ ボックスが表示されます。
- d) 次のいずれか 1 つのステップを行います：
 - 基本状態を使用するには、**基本状態の実行** チェック ボックスをオンにします。
 - 基本状態をオフにするには、**基本状態の実行** チェック ボックスをオフにします。
- e) **OK** をクリックします。

2. スクリプトで、以下のステップを実行します。

- a) **プロパティ** ペインで、変更するアプリケーション構成を右クリックし、**アプリケーションの実行可能ファイル名の編集** を選択します。
たとえば、Internet Explorer で Web アプリケーションをテストする場合は、メニューから **ieexplore.exe の編集** を選択します。**アプリケーション構成の編集** ダイアログ ボックスが表示されます。
- b) 次のいずれか 1 つのステップを行います：
 - 基本状態を使用するには、**基本状態の実行** チェック ボックスをオンにします。
 - 基本状態をオフにするには、**基本状態の実行** チェック ボックスをオフにします。
- c) **OK** をクリックします。

基本状態の実行

アプリケーションに対してテストの記録を開始する前に基本状態を実行することで、記録するすべてのアプリケーションを起動して、記録に適した状態にすることができます。

基本状態を実行すると、アクティブなビジュアル テストまたは .NET スクリプトのすべてのアプリケーション構成のうち、**基本状態の実行** が設定されているものが実行されます。

基本状態を実行するには：

操作 > 基本状態の実行 をクリックします。

アプリケーションの種類に応じて、Silk Test Workbench は次のアクションを実行します。

- 現在のプロジェクトで定義されているアプリケーション構成に対応するすべてのアプリケーションのアプリケーション構成を実行します。

- デスクトップ アプリケーションの場合、Silk Test Workbench は指定したインストール ディレクトリからアプリケーションを開きます。%ProgramFiles% 環境変数を含むディレクトリへのパスが指定された場合、Silk Test Workbench は Program Files ディレクトリからアプリケーションを検索し、見つからなかった場合に、Silk Test Workbench は Program Files (x86) ディレクトリを検索します。%ProgramFiles(x86)% 環境変数を含むディレクトリへのパスが指定された場合、Silk Test Workbench は Program Files (x86) ディレクトリからアプリケーションを検索し、見つからなかった場合に、Silk Test Workbench は Program Files ディレクトリを検索します。
- Web アプリケーションの場合、Silk Test Workbench は指定したブラウザで指定した URL の Web アプリケーションを開きます。
- モバイル Web アプリケーションの場合、Silk Test Workbench は指定したモバイル デバイスまたはエミュレータ上の指定したブラウザで指定した URL を開きます。
- モバイル ネイティブ アプリケーションの場合、Silk Test Workbench は指定したモバイル デバイスまたはエミュレータ上で指定したアプリを開きます。指定したアプリが指定したモバイル デバイスまたはエミュレータ上にインストールされていない場合、Silk Test Workbench はアプリをインストールしてから開きます。

ユーザー インターフェイスからの基本状態の変更


Silk Test Workbench が記録や再生時にテスト対象アプリケーションを開始する方法を指定する基本状態を、ユーザー インターフェイスから編集できます。基本状態では、AUT の実行可能ファイルの場所、作業ディレクトリ、Web アプリケーションの URL や接続文字列などを指定できます。たとえば、ステージング Web サイトで既に実行したテストを、プロダクション Web サイトで実行する場合には、基本状態の URL を変更すれば、新しい Web サイトに対してテストを実行することができます。



注: 再生時に特定のテストに対してのみ Silk Test Workbench がテスト対象アプリケーション (AUT) を開始するように指定する場合は、テストを含むスクリプトで基本状態を編集します。詳細については、「[スクリプトでの基本状態の変更](#)」を参照してください。


ユーザー インターフェイスから基本状態を編集するには：

1. プロジェクト全体に関連付けられているアプリケーション構成を変更するには：
 - a) Silk Test Workbench メニューから **ツールアプリケーション構成の編集** を選択します。 **プロジェクトの選択** ダイアログが開きます。
 - b) リストからプロジェクトを選択します。
 - c) **編集** をクリックします。 **アプリケーション構成の編集** ダイアログが開きます。
 - d) 該当するアプリケーション構成の隣にある **編集** をクリックします。
2. 特定のビジュアル テストに関連付けられているアプリケーション構成を変更するには：
 - a) ビジュアル テストを開きます。
 - b) **タスク** ペインで **<<開始>>** ステップをクリックします。
 - c) **プロパティ** ペインの **アプリケーション構成** カテゴリに移動します。
 - d) 変更するアプリケーション構成をクリックします。
 - e) **アプリケーション構成の編集** をクリックします。
3. 特定の VB .NET スクリプトに関連付けられているアプリケーション構成を変更するには：
 - a) VB .NET スクリプトを開きます。
 - b) **プロパティ** ペインの **アプリケーション構成** カテゴリに移動します。
 - c) 変更するアプリケーション構成をダブルクリックします。
4. **基本状態の実行** チェック ボックスをオンにし、アプリケーション構成で基本状態を使用します。
5. Web アプリケーションまたはモバイル Web アプリケーションをテストするには、リストからインストール済みのブラウザまたはモバイル ブラウザのうちの 1 つを選択します。
変更 をクリックして **アプリケーションの選択** ダイアログ ボックスを開き、使用するブラウザを選択することもできます。
6. 実行可能ファイルを指定するには、**実行可能ファイル** フィールドに実行可能ファイルへの完全パスを入力します。

 **注:** Web アプリケーションをテストする場合に、ブラウザの実行可能ファイルを指定する場合は、ブラウザの種類からカスタムを選択します。

たとえば、Mozilla Firefox を起動するには、「C:\Program Files (x86)\Mozilla Firefox\firefox.exe」を入力します。

7. 実行可能ファイルパターンを指定して、デスクトップ アプリケーションをテストする場合は、**実行可能ファイルパターン** テキストボックスに、テストするデスクトップ アプリケーションの実行可能ファイルの名前とファイルへのパスを入力します。
たとえば、電卓を指定する場合には、*calc.exe と入力します。
8. デスクトップ アプリケーションをテストし、実行可能ファイルと一緒にコマンド ラインパターンを使用したい場合には、コマンド ラインパターンを **コマンド ラインパターン** テキストボックスに入力します。
9. コマンド ライン引数を指定するには、**コマンド ライン引数** フィールドに引数を入力します。

 **注:** Web アプリケーションをテストする場合に、コマンド ライン引数を指定してブラウザを開始する場合は、ブラウザの種類からカスタムを選択します。

たとえば、Mozilla Firefox を *myProfile* プロファイルを指定して起動するには、「-p myProfile」を入力します。

10. テストするアプリケーションがディレクトリに依存する場合は、**作業ディレクトリ** フィールドにディレクトリへのパスを指定します。
たとえば、Java アプリケーションを起動するバッチ ファイルを使用する場合には、バッチ ファイルは JAR ファイルを相対パスで参照することができます。この場合、正しく相対パスが処理されるように作業ディレクトリを指定します。
11. デスクトップ アプリケーションをテストする場合は、**ロケーター** フィールドにアプリケーションのメイン ウィンドウを指定します。
たとえば、ロケーターは、/Shell[@caption='SwT Test Application'] のようになります。

12. Web アプリケーションをテストする場合は、**移動する URL** テキストボックスに、Web アプリケーションのアドレスを入力します。

13. 省略可能 : あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。

14. 省略可能 : ブラウザー ウィンドウの **向き** を選択します。

15. 省略可能 : **ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。

16. リモート ロケーション上の Web アプリケーションまたはモバイル ネイティブ アプリケーション (Mac に接続されているモバイル デバイスなど) をテストする場合は、リモート ロケーションを変更する場合は、**変更** をクリックして **アプリケーションの選択** ダイアログ ボックスを開き、**リモート ロケーションの編集** をクリックします。

17. モバイル アプリケーションや Apple Safari 上の Web アプリケーションをテストする場合は、**接続文字列** テキストボックスに接続文字列を入力します。

詳細については、「[接続文字列](#)」を参照してください。

18. WebDriver ベースのブラウザのキーパビリティを編集する場合は、**接続文字列** テキストボックスを使用できます。

たとえば、Google Chrome のブラウザ ウィンドウを最大化して開始する場合には、**接続文字列** テキストボックスに次のように入力します。

```
chromeOptions={"args":["--start-maximized"]}
```

詳細については、「[WebDriver ベースのブラウザのキーパビリティの設定](#)」を参照してください。

19. モバイル ネイティブ アプリケーションをテストしている場合は、アプリケーションを指定します。

- モバイル デバイスまたはエミュレータ上にアプリをインストールする場合は、**参照** をクリックしてアプリ ファイルを選択するか、**アプリ ファイル** テキスト フィールドにアプリ ファイルへの完全パ

スを入力します。このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。


- Android デバイス上に既にインストールされているアプリを使用する場合は、**パッケージ/アクティビティ** リストからアプリを選択するか、**パッケージ/アクティビティ** フィールドにパッケージとアクティビティを指定します。
- iOS デバイス上に既にインストールされているアプリを使用する場合は、**バンドル ID** を指定します。
- UFT Mobile 上で利用可能なアプリを使用する場合は、**アプリ識別子** を指定します。

20OK をクリックします。

21 テスト対象アプリケーションの起動に時間がかかる場合は、再生オプションのアプリケーション準備完了タイムアウトの値を増やしてください。


アプリケーションが実行していない場合は、基本状態を実行することで開始されます。アプリケーションがすでに実行中の場合には、Silk Test Workbench によってアプリケーションの別のインスタンスは開始されません。

複数のアプリケーション構成がテストに含まれており、基本状態に関連付けられているオブジェクト以外のアプリケーションまたは Web ページを変更する場合は、基本状態をオフにすることができます。このことは、変更を記録したり再生したりするときに基本状態が使用されないことを意味します。このため、アプリケーションまたは Web ページを起動するステップをテスト内に記録する必要があります。たとえば、Web ページをテストする場合は、テスト内で Internet Explorer を起動します。

 **注:** 定義済みの基本状態を使って Web アプリケーションをテストするときに、ブラウザ アプリケーション構成を複数追加しないでください。

スクリプトでの基本状態の変更

Silk Test Workbench が再生時にテスト対象アプリケーションを開始する方法を指定する基本状態を、スクリプトで編集できます。基本状態では、AUT の実行可能ファイルの場所、作業ディレクトリ、Web アプリケーションの URL や接続文字列などを指定できます。たとえば、ステージング Web サイトで既に行ったテストを、プロダクション Web サイトで実行する場合には、基本状態の URL を変更すれば、新しい Web サイトに対してテストを実行することができます。

 **注:** Silk Test Workbench が記録や再生時にテスト対象アプリケーション (AUT) を開始する方法を指定するには、ユーザー インターフェイスから基本状態を編集します。詳細については、「ユーザー インターフェイスからの基本状態の変更」を参照してください。

スクリプトで基本状態を編集するには：

1. スクリプトを開きます。
2. `baseState` メソッドを変更します。

基本状態の作成と実行との間にコードを追加できます。

```
' VB code
' Go to web page 'demo.borland.com/InsuranceWebExtJS'
BrowserBaseState baseState = new BrowserBaseState()
' <-- Insert your changes here!
baseState.Execute()
```

3. テストするアプリケーションの実行可能ファイルの名前とファイルへのパスを指定する場合は、次のコードを使用します。

```
' VB code
baseState.Executable=executable
```

たとえば、電卓を指定する場合は、次のように入力します。

```
' VB code
baseState.Executable = "C:¥¥Windows¥¥SysWOW64¥¥calc.exe"
```

Mozilla Firefox を指定する場合は、次のように入力します。

```
' VB code
baseState.Executable = "C:¥¥Program Files (x86)¥¥Mozilla Firefox¥¥firefox.exe"
```

4. コマンドライン引数を指定する場合は、次のコードを使用します。

```
' VB code
baseState.CommandLineArguments = commandLineArguments
```

たとえば、Mozilla Firefox を *myProfile* プロファイルを指定して起動する場合は、次のように入力します。

```
' VB code
baseState.CommandLineArguments = "-p myProfile"
```

5. 他の作業ディレクトリを指定する場合は、次のコードを使用します。

```
' VB code
baseState.WorkingDirectory = workingDirectory
```

6. 実行可能ファイルパターンを使用する場合には、次のコードを使用します。

```
' VB code
baseState.ExecutablePattern = executablePattern
```

たとえば、電卓に対して実行可能ファイルパターンを指定する場合は、次のように入力します。

```
' VB code
baseState.ExecutablePattern = "*¥¥calc.exe"
```

7. 実行可能ファイルと一緒にコマンドラインパターンを使用する場合は、次のコードを使用します。

```
' VB code
baseState.CommandLinePattern = commandLinePattern
```

たとえば、アプリケーションのコマンドラインが **com.example.MyMainClass** で終わる場合には、コマンドラインパターンに ***com.example.MyMainClass** を使用します。

```
' VB code
baseState.CommandLinePattern = "*com.example.MyMainClass"
```

8. Web アプリケーションまたはモバイル Web アプリケーションをテストする場合、現在のプロジェクトに対してアプリケーション構成が設定されていない場合は、インストール済みのブラウザまたはモバイルブラウザから 1 つを指定します。

たとえば、Google Chrome を指定する場合は、次のように入力します。

```
' VB code
baseState.BrowserType = BrowserType.GoogleChrome
```

9. Web アプリケーションまたはモバイル Web アプリケーションをテストする場合、現在のプロジェクトに対してアプリケーション構成が設定されていない場合は、テストする Web アプリケーションのアドレスを指定します。

```
' VB code
baseState.Url = url
```

たとえば、次のように入力します。

```
' VB code
baseState.Url = "demo.borland.com/InsuranceWebExtJS/"
```

10. デスクトップブラウザ上の Web アプリケーションをテストする場合は、ブラウザ ウィンドウの幅と高さを指定できます。

```
' VB code
baseState.ViewportHeight = viewportHeight
baseState.ViewportWidth = viewportWidth
```

11. リモートロケーション上の Web アプリケーション、またはモバイルネイティブアプリケーションをテストする場合は、接続文字列を指定します。

```
' VB code
New MobileBaseState(connectionString)
```

接続文字列についての詳細は、「[リモートデスクトップブラウザの接続文字列](#)」または「[モバイルデバイスの接続文字列](#)」を参照してください。

12 Mozilla Firefox または Google Chrome のケイパビリティを編集する場合にも、接続文字列を使用できません。

たとえば、Mozilla Firefox のダウンロード フォルダーを設定する場合は、次のように入力します。

```
' VB code
baseState.ConnectionString = "moz:firefoxOptions= {¥"prefs¥": { ¥"browser.download.dir
¥":¥"C:/Download/¥"}};"
```

詳細については、「[WebDriver ベースのブラウザのケイパビリティの設定](#)」を参照してください。

記録の概要

新しいビジュアルテストまたはスクリプトを記録すると、**アプリケーションの選択** ダイアログ ボックスが開きます。テストするアプリケーションをダブルクリックします。Silk Test Workbench により、テストアプリケーションを識別するアプリケーション構成と基本状態が自動的に作成されます。アプリケーション構成は、テストするアプリケーションに Silk Test Workbench が接続し、構成する方法を定義します。アプリケーションの基本状態とは、各テストの実行開始前にアプリケーションに想定される既知の安定した状態です。

ビジュアルテストまたはスクリプトを記録する前に **オブジェクトの識別** ダイアログ ボックスを開いた場合にも、**アプリケーションの選択** ダイアログ ボックスが開きます。

テストするアプリケーションを選択したら、以下のステップのいずれかを実行します。

- ビジュアルテストを記録する。
- スクリプトを記録する。
- 識別するオブジェクトを選択する。

必要に応じて、アプリケーション構成を変更し、基本状態をオフにすることができます。

記録中のオブジェクトのハイライト

記録中に AUT のアクティブ オブジェクトは緑色の矩形でハイライトされます。新しいオブジェクトがアクティブになるとすぐに、この新しいオブジェクトがハイライトされます。同じオブジェクトがアクティブなまま 0.5 秒が経過すると、ツールチップが表示されます。ツールチップには、アクティブ オブジェクトのクラス名とアクティブ オブジェクトに相対的なマウスの現在位置が表示されます。このツールチップは、新しいオブジェクトがアクティブになったり、ユーザーがマウスを押したり、表示されてから 2 秒が経過すると、表示されなくなります。

記録中および再生中に除外される文字

記録および再生中に Silk Test が無視する文字を以下に示します。

文字	コントロール
...	MenuItem
タブ	MenuItem
&	すべてのコントロール。アンパサンド (&) はアクセラレータとして使用されるため、記録されません。

記録中に利用可能なアクション

記録中に次のアクションを **記録中** ウィンドウで実行できます。

アクション	ステップ
記録の一時停止	一時停止 をクリックして、操作を記録せずに AUT を特定の状態にしてから、 記録 をクリックして記録を再開できます。
記録した操作の順番の変更	記録中 ウィンドウで記録した操作の順番を変更するには、移動したい操作を選択して新しい場所にそれらをドラッグします。
複数の操作の選択	複数の操作を選択するには、Ctrl を押しながら操作をクリックするか、もしくは選択する最初の操作をクリックしてから、 Shift を押しながら最後のアクションをクリックします。
記録した操作の再生	記録中 ウィンドウから記録した操作を再生するには、操作を選択してから 再生 をクリックします。すべての記録した操作を再生するには、 記録した操作 をクリックしてから 再生 をクリックします。
記録した操作の削除	誤って記録した操作を 記録中 ウィンドウから削除するには、マウス カーソルをその操作上にポイントすると表示される 削除 をクリックします。
イメージまたはコントロールのプロパティの検証	検証するオブジェクトの上にマウス カーソルを移動して、 Ctrl+Alt を押しします。
オブジェクト マップ エントリの変更	Web アプリケーション、またはモバイル Web アプリを記録しているときに、記録したオブジェクトに対して自動的に生成されたオブジェクト マップ エントリが分かりにくい場合や、特殊な文字が含まれてしまっている場合、分かりやすいオブジェクト マップ エントリに変更したいことがあります。このような場合は、記録中にオブジェクトを右クリックして、 操作の選択 ダイアログの オブジェクトの識別 領域を展開します。そして、 オブジェクト マップ ID フィールドでオブジェクト マップ エントリを編集できます。たとえば、デモアプリケーションの画像に対して自動的に生成されるオブジェクト マップ エントリは、「 <i>http demo borland</i> 」になります。このエントリが何を参照しているのかを理解するのは、オブジェクト マップを見ただけでは困難です。このオブジェクト マップ エントリを「 <i>InsuranceWebHomePageBanner</i> 」のように変更すれば、より分かりやすくなります。この機能は、Internet Explorer 以外のすべてのサポートするデスクトップおよびモバイル ブラウザで利用できます。
別のロケータの選択	Web アプリケーション、またはモバイル Web アプリに対して記録しているときに、記録したオブジェクトに対して自動的に生成されたロケータが要求を満たしていない場合、 ロケータ フィールドにある矢印をクリックして、Silk Test Workbench に他のロケータを提示させることができます。提示されるロケータはすべて、オブジェクトを一意に識別できるものです。この機能は、Internet Explorer 以外のすべてのサポートするデスクトップおよびモバイル ブラウザで利用できます。

テストの拡張

ビジュアルテストまたはスクリプトを拡張するさまざまな方法について説明します。

アクティブ データ


ビジュアルテストとスクリプトには、通常、定数またはリテラルデータの値が含まれていて、アプリケーションのオープン フィールドや複数選択のコントロールに自動的にデータが入力されます。このようなアプリケーションをテストする場合、ビジュアルテストまたはスクリプトでは、各トランザクションに対し同じデータ フィールド名およびデータ値が使用されます。

リテラルデータ以外のデータを使用してビジュアルテストやスクリプトを繰り返し再生するには、別のデータ値が使用されるようにビジュアルテストやスクリプトを変更する必要があります。

アクティブ データ テストでは、強力で包括的なアプリケーション テスト ソリューションの入力として既存のデータを利用できます。アクティブ データを使用して、トランザクションごとに異なるデータ セットを使用することにより、複雑なコードを記述したり、既存のデータを損なうことなく、テスト アプリケーションで複数のトランザクションを実行できます。

既存のデータ ファイルを手動で編集することなく、既存のデータ ファイルに新しいデータを書き込むこともできます。

アクティブ データを使用すると、テスト担当者は、有効なデータや無効なデータをテスト アプリケーションに入力して、その結果を検証できます。つまり、一連のテスト条件を実行するためのビジュアルテストやスクリプトの数が少なくて済み、メンテナンスのためのオーバーヘッドが減ります。

 **注:** アクティブ データ ファイルを開くときにエラーが発生した場合は、エラー処理を使用してテスト実行を続行できます。

ビジュアル テスト

ビジュアルテストでは、ローカル変数を使用する場所にアクティブ データを使用できます。入力をアクティブ データ ファイルのデータに置き換えたり、一連の入力ステップを繰り返す繰り返しロジックを作成し、記録されたデータをアクティブ データ ファイルのデータに置き換えたりできます。式の内容など、ビジュアルテストで作成されたデータはすべて、アクティブ データ ファイルに書き込んでテストに利用することもできます。

ビジュアルテストで使用するアクティブ データ ファイルを選択するには、ビジュアルテストの <<開始 >> ステップのアクティブ データ プロパティで設定するか、または任意のステップを選択して **挿入 > アクティブ データ** を選択します。

スクリプト

スクリプトでは、ローカル変数を使用する場所にアクティブ データを挿入できます。ただし、ビジュアルテストとは異なり、ウィザードは使用できません。手動でデータを挿入する必要があります。

キーワード駆動テスト

キーワード駆動テストでは、テスト全体に対してアクティブ データを簡単に使用することはできません。しかし、個々のキーワードを実装しているビジュアルテストや .NET スクリプト内では、アクティブ データを使用することができます。このようにキーワード内でアクティブ データを使用するには、アクティブ データを使用したキーワードの種類の実装について説明するトピックを参照してください。

アクティブ データ テストの利点

アプリケーション テストでは、テスト対象アプリケーション (AUT) のコントロールにデータを入力して、そのデータを処理し、最終的に結果のレポートを得ることで、AUT が期待通り機能していることを確認します。テストの記録時に入力したデータを変更せずに、テストを再生すると、ある特定のデータ セットに対して AUT が機能することだけを検証することになります。

サポートするシナリオに対して個々にテストを記録するのではなく、エッジケースを含む様々なシナリオに対して AUT をテストできるようなテストを作成できるようにするために、Silk Test Workbench にはアクティブ データテスト機能が備わっており、次のメリットを提供します。

- テストの実行時に様々なデータ セットを使って、単一のテストで様々なサポートするシナリオやエッジケースをテストする
- 実際のテストを変更することなく、テストデータをアクティブ データ ファイルで直接変更したり、Silk Test Workbench 内から変更する


アクティブ データ資産

アクティブ データは、Excel スプレッドシートのようなファイルに格納され、これをアクティブ データ ファイルと呼びます。アクティブ データ ファイルは、Silk Test Workbench データベース内部に格納することも、ファイル システムなどの外部に格納することも可能です。

アクティブ データ資産を使うと、アクティブ データ ファイル内のデータに Silk Test Workbench からアクセスできます。また、アクティブ データ資産を使って、Silk Test Workbench から直接アクティブ データ ファイル内のデータを管理することもできます。アクティブ データ資産は、ビジュアル テスト、VB .NET スクリプト、キーワード駆動テストのすべてから使用することができ、いつでも編集できます。

アクティブ データ資産を作成する場合、資産で使用する外部ファイルを選択します。次の項目を指定できます。

- アクティブ データ資産内で使用するアクティブ データ ファイルのデータ
- アクティブ データ ファイルの最初の行が列ヘッダーかどうか
- アクティブ データ ファイルを書き込み可能にするか、読み取り専用にするか


 **注:** Silk Test Workbench が開いているアクティブ データ ファイルにアクセスしようとする時、実行時エラーが発生します。このような場合にテスト実行を続行するには、エラー処理を使用します。

アクティブ データ ファイルは事前に作成しておくことも、Silk Test Workbench 内で作成して編集することも可能です。アクティブ データ資産は、**アセット ブラウザ** を使用して作成、変更、および管理します。

アクティブ データ テスト資産の作成

新しいアクティブ データ資産を作成するには、以下を実行します。

1. **表示 > アセット ブラウザ** を選択します。
2. **資産の種類** で、**アクティブ データ** を右クリックして、**アクティブ データの新規作成** を選択します。**アクティブ データ** ウィンドウが開きます。
3. **名前** テキスト ボックスに、資産のわかりやすい名前を入力します。
アクティブ データ資産名は、Silk Test Workbench 資産の命名規則に従う必要があります。

 **ヒント:** アクティブ データ資産に、その資産に使用されるアクティブ データ ファイルと同じ名前を付けると、テストで使用する資産を特定しやすくなります。

4. 資産が使用するデータを含んだ新しいアクティブ データ ファイルを作成するか、既存のアクティブ データ ファイルを選択できます。
 - 新しいアクティブ データ ファイルをファイル システムに作成するには、**外部ファイル** を選択してから **新規作成** をクリックします。

- 既存のアクティブ データ ファイルのデータを使用するには、**外部ファイル** を選択してから **参照** をクリックし、ファイルを選択します。アクセスできるようにするため、ファイルを閉じてください。
- 新しいアクティブ データ資産を Silk Test Workbench データベースに作成するには、**内部データ** を選択してから **新規作成** をクリックします。
- 既存のアクティブ データ ファイルを Silk Test Workbench データベースに格納して、アクティブ データ資産を使って内部ファイルからデータを使用するには、**内部データ** を選択してから **アップロード** をクリックし、ファイルを選択します。
- 既存のアクティブ データ ファイルを Silk Test Workbench データベースからファイル システムにコピーするには、**内部データ** を選択してから **ダウンロード** をクリックし、ファイルを選択します。保存したファイルは、Excel などのアプリケーションを使って編集できます。

ファイル テキスト ボックスには、アクティブ データ ファイルの名前が表示されます。

5. 省略可能：説明 テキスト ボックスに、資産の説明を入力します。

説明を使用して、データ ファイルとして使用されるファイルの種類、資産を作成するアクティブ データ テストによって実行されるテストの種類、またはアクティブ データ テストでテストされるアプリケーションを示すことができます。**アセットブラウザ** では、説明が入力されているすべてのアクティブ データ資産に説明が表示されます。



注: 空のデータ ファイルが必要な場合は、**保存** および **閉じる** をクリックしてすべての変更内容を保存し、**アクティブ データ** 設定ウィンドウを閉じます。

6. アクティブ データ ウィンドウで、**オプション** タブに切り替えます。

7. アクティブ データ ファイルに対して読み取りオプションを設定します。

8. アクティブ データ ウィンドウで、**詳細** タブに切り替えます。

9. アクティブ データ ファイル内のデータを編集します。

10. **保存** または **データの保存** をクリックしてすべての変更内容を保存するか、または **保存して閉じる** をクリックしてすべての変更内容を保存し、**アクティブ データ** ウィンドウを閉じます。

アクティブ データ資産のデータ ファイルが外部の場合、つまりファイル システムに格納されており、Silk Test Workbench データベースではない場合、**データの保存** をクリックすると、データ ファイルだけが保存されます。この場合、アクティブ データ資産は保存されません。

新しいアクティブ データ資産が **アセットブラウザ** に表示されます。

アクティブ データ テストのためのテスト ファイルの準備

Silk Test Workbench では、アクティブ データ テストで以下の種類のファイルを使用できます。

- Microsoft Excel® スプレッドシート (.xls) ファイル



注: Microsoft Excel® がインストールされていない場合、読み取り専用モードで .xls ファイルを使用できます。

- Microsoft Excel® スプレッドシート (.xlsx) ファイル



注: Silk Test Workbench で .xlsx ファイルを使用するには、Microsoft Excel® 2007 以降をインストールしてください。Microsoft Excel® 2007 以降がインストールされていない場合、**2007 Office System** ドライバ：データ接続コンポーネントをインストールして、読み取り専用モードで .xlsx ファイルを使用できます。

- プレーン テキスト ファイル (.txt)
- カンマ区切り値ファイル (.csv)

カンマ区切り値ファイルおよびプレーン テキスト ファイルには、1 セットのデータしか含めることができませんが、Excel スプレッドシート ファイルには複数のシートを含めることができます。各シートには独自の表が含まれ、アクティブ データ テストでは別のエンティティとして扱われます。

カンマ区切り値ファイルまたはプレーン テキスト ファイルを使用して新しいテスト データ資産を作成する場合は、ファイルだけを選択すれば済みます。Excel スプレッドシート ファイルを使用して新しいテスト データ資産を作成する場合は、使用するシートも選択する必要があります。

アクティブ データ テストのためのファイル準備のヒント

- 可能な場合、データ ファイルのデータは、アプリケーション内と同じデータ カテゴリにグループ化する必要があります。たとえば、データ ファイルにアクティブ データとして使用される姓と名が含まれている場合、テスト対象アプリケーションには姓と名の別個のフィールドがなければなりません。アプリケーションで姓と名に別のフィールドが使用され、データ ファイルでは姓と名を組み合わせることで 1 つのデータ フィールドになっている場合、テスト データと、アプリケーションのテストに必要なデータが 1 対 1 の関係になりません。アクティブ データ ファイル内の結合されたデータをテスト アプリケーションの姓フィールドまたは名フィールドのテストに使用した場合は、期待した結果が得られない場合があります。
- アクティブ データ テストに使用するファイルは、ビジュアル テストまたはスクリプトで使用する前に作成しておく必要があります。
- オブジェクト名テスト データの場合、データ ファイル内の名前エントリにはかっこや引用符を含めないでください。

たとえば、あるフォーム上の複数のエディット ボックスで同じテストを実行し、それらの各エディット ボックスは Index プロパティのみを使用して一意に識別される場合、各エディット ボックスの名前は以下のように表されます。

```
TextField("[1]")
TextField("[2]")
TextField("[3]")
TextField("[4]")
```

アクティブ データ ファイル内のオブジェクト名のテスト データの各行には、フォーム上のエディット ボックスを表すそれぞれのインデックスを含めることができます。ただし、アクティブ データ ファイルの各行のデータでは、以下のようにかっこまたは引用符を除いてください。

```
Index=1
Index=2
Index=3
Index=4
```

- アクティブ データ ファイルの最初の行は、データの最初の行にすることも列見出しにすることもできます。アクティブ データ テストを設定するとき、アクティブ データ資産では、アクティブ データ ファイルの最初の行のタイプを適切に設定する必要があります。
- ユーザーがアクティブ データ資産を読み取り専用として開くと、読み取り権限だけが付与されます。アクティブ データ資産が読み書き可能モードで開かれると、資産に一度も書き込みが行われなくても、読み取りと書き込みの権限が付与されます。これは、テストがあとでアクティブ データ ファイルに書き込みをするように変更された場合でも、正しく再生されるようにするためです。読み取りおよび書き込み権限で開かれたアクティブ データ資産は、排他的ファイル ロック機能を持ち、資産と関連付けられたアクティブ データ ファイルが他のプログラムから開かれられないようにすることができます。

アクティブ データ テスト用の新しいデータ ファイルの作成

アクティブ データ資産には、一緒に使用するデータ ファイルを関連付ける必要があります。Silk Test Workbench を使用してファイルを作成するか、またはテキスト エディタや Microsoft Excel® などのツールを使用してファイルを作成します。

アクティブ データ資産の作成時に、Silk Test Workbench で、資産に関連付けるアクティブ データを含む新しいアクティブ データ ファイルを作成することもできます。

1. **表示 > アセット ブラウザ** を選択します。
2. **資産の種類** で、**アクティブ データ** を右クリックして、**アクティブ データの新規作成** を選択します。 **アクティブ データ** ウィンドウが開きます。
3. **名前** テキスト ボックスに、資産のわかりやすい名前を入力します。

アクティブ データ資産名は、Silk Test Workbench 資産の命名規則に従う必要があります。



ヒント: アクティブ データ資産に、その資産に使用されるアクティブ データ ファイルと同じ名前を付けると、テストで使用する資産を特定しやすくなります。

4. **新規作成** をクリックして、アクティブ データ ファイルとして使用する新しいファイルを作成します。**アクティブ データの新規作成** ダイアログ ボックスが開きます。
5. **名前** テキスト ボックスに、新しいファイルのわかりやすい名前を入力します。
このファイルに含まれるデータを示す名前か、このファイルを使用するテストの対象アプリケーションの種類を示す名前を付けることをお勧めします。
6. 新しいアクティブ データ ファイルの種類を選択します。
ファイルの種類の詳細については、「アクティブ データ テスト資産の作成」を参照してください。
7. **参照** をクリックし、**フォルダの参照** ダイアログ ボックスを使用して、新しいアクティブ データ ファイルを保存する場所を選択します。**OK** をクリックして、**アクティブ データの新規作成** ダイアログ ボックスに戻ります。新しいアクティブ データ ファイルの場所が、**場所** テキスト ボックスに表示されます。
8. **OK** をクリックして、新しいアクティブ データ ファイルを作成します。
9. アクティブ データ資産の作成を続行します。


アクティブ データ テスト ファイルに対する読み取りオプションの設定

アクティブ データ 設定ウィンドウの **オプション** タブを使用して、アクティブ データ ファイルに対する読み取り/書き込みオプションを設定します。また、このタブでは、アクティブ データ ファイルの最初の行を列見出しとして扱うかどうかを設定したり、アクティブ データ テストに使用するスプレッドシート データ ファイル中のシートを設定したりもします。これらのオプションの設定手順は、スプレッドシート データ ファイルに対するものとテキスト (.txt) またはカンマ区切り (.csv) データ ファイルに対するものとで多少異なります。

1. Silk Test Workbench によってファイルの最初のデータ行が列見出しとして扱われるようにするには、**アクティブ データ** ウィンドウの **オプション** タブで **最初の行を見出しとして使用** チェック ボックスをオンにします。
このチェック ボックスをオンにすると、アクティブ データ テストの作成時に、データ ファイル内の各列の行に含まれるレコードに対してデータをマップできます。**最初の行を見出しとして使用** チェック ボックスがオフの場合、ファイルの最初の行のデータは実際のアクティブ データ項目として解釈されます。
2. ファイルが実際に読み取り専用設定されているかどうかにかかわらず、Silk Test Workbench がアクティブ データ ファイルを読み取り専用ファイルとして扱うよう強制するには、**読み取り専用を適用** チェック ボックスをオンにします。
このオプションを選択しても、アクティブ データ ファイルの読み取り/書き込み設定は変更されません。
読み取り専用を適用 チェック ボックスをオンにすると、**アクティブ データ** ウィンドウの **詳細** タブの行と列の **挿入** および **削除** ボタンも無効になります。そのため、Silk Test Workbench 内からアクティブ データ ファイルを変更できなくなります。
3. スプレッドシートでは、**シート** リストで、アクティブ データ資産で使用するデータを含む Excel スプレッドシート内のシートを選択します。
4. カンマ区切り値ファイルおよびテキスト ファイルでは、アクティブ データ ファイル内でデータ項目を区切るために使用される文字を指定するオプションを選択します。
カスタム文字を使用する場合は、**その他** オプションを選択して、**その他** テキスト ボックスにカスタム文字を入力します。
5. アクティブ データ資産の作成を続行します。


アクティブ データ テストのためのアクティブ データ ファイルの編集


アクティブ データ ウィンドウの **詳細** タブを使用して、アクティブ データ資産で使用されるアクティブ データ ファイル内のデータを編集します。 **全般** タブでアクティブ データ ファイルの名前が指定されるまでは、**詳細** タブの機能は無効になっています。 **読み取り専用を適用** チェックボックスがオフの場合は、列と行をデータ ファイルに追加したり、データ ファイルから削除したりできます。また、データをリストからコピーしたり、リストに貼り付けたりすることもできます。

 **ヒント:** ファイルの元のアプリケーションでアクティブ データ ファイルを編集することもできます。元のアプリケーションを使用してアクティブ データ ファイルを変更した場合、その変更内容は Silk Test Workbench で次回そのファイルを開いたときに表示されます。

1. アクティブ データ ファイルの内容を必要に応じて更新します。
 - a) 該当するボタンをクリックして行または列を挿入します。行と列は、最後にある行または列のあとに自動的に挿入されます。
 - b) 行または列内のセルを 1 つ選択し、該当する **削除** ボタンをクリックして行または列を削除します。次に、メッセージ ボックスで **はい** をクリックして削除を確認します。その行または列内のすべてのセルの、すべてのデータも削除されます。
 - c) データを入力または編集するセル内をダブルクリックします。データは入力することも、外部ソースからコピーしてセルへ貼り付けることもできます。
 - d) 列のセルを選択し、**名前の変更** をクリックして、その列の見出しの名前を変更します。
 - e) 任意の時点で **データの保存** をクリックして、すべての更新内容を保存します。 **データの保存** をクリックすると、変更内容が元のアクティブ データ ファイルに保存されます。特定のテスト実行に対してのみデータを変更する場合は、最初に元のアクティブ データ ファイルのコピーを作成します。
2. アクティブ データ資産の作成を続行します。

再生中に、アクティブ データ テストに使用されるアクティブ データ ファイルのデータを更新することができます。 ビジュアル テストの場合は、更新内容が元のアクティブ データ ファイルに自動的に保存されますが、データの更新を元のアクティブ データ ファイルに保存する方法を制御するステップを作成することもできます。詳細については、「**詳細設定オプションの変更**」を参照してください。 スクリプトの場合は、再生中に .NET コマンドを使用してアクティブ データ ファイルを編集することができます。

 **ヒント:** **詳細** タブを使用してアクティブ データ ファイルのデータを更新して保存すると、Silk Test Workbench によって対応するアクティブ データ資産のバージョン番号が更新されます。ただし、アクティブ データ ファイル内のデータを Microsoft Excel などの外部ツールを使用して更新すると、資産のバージョン番号は更新されません。

 **ヒント:** データは、アクティブ データ テストに必要なとされる引用符または後続スペースを含めて追加する必要があります。

アクティブ データに使用するデータの指定

アクティブ データ テストにアクティブ データ ファイルのデータを使用する場合、使用するファイル内のデータ行および使用順序を指定することができます。 **開始行**、**終了行** および **ランダム数** の値によって、使用される行および行の使用順序が決定されます。

ランダム数 に指定する値により、資産に関連付けられているアクティブ データ ファイルのレコードをアクティブ データ テストで使用するかどうか、またどのように使用するかが決まります。

ビジュアル テストでは、**開始行**、**終了行**、および **ランダム数** の値は、**テスト ロジック (繰り返し) デザイン** の **使用するアクティブ データ資産の定義** ページで設定します。これらの値は、設定後、「アクティブ データを使用して繰り返します」ステップの **開始行**、**終了行**、および **ランダム数** プロパティで更新できます。VB .NET テスト スクリプトの場合、**開始行**、**終了行**、および **ランダム数** の値を LoadActiveData メソッドの StartRow、EndRow、および RandomCount パラメータを使って設定および更新します。

以下の表は、さまざまな **開始行** と **終了行** の値を許容される **ランダム数** の値と共に使用した結果を示しています。

[開始行]の値	[終了行]の値	[ランダム数]の値 (ビジュアルテスト)	[ランダム数]の値 (VB .NET スクリプト)	結果
1	-1 ^b	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	すべての行を、インデックスの順序で読み込みます。
1	5	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	行を 1、2、3、4、5 の順で読み込みます。
5	-1 ^b	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	アクティブ データ ファイルの行 5 から最後の行までを順番に読み込みます。
5	1	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	行を 5、4、3、2、1 の順で読み込みます。
-1 ^a	1	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	すべての行を、逆の順序で読み込みます。
-1 ^a	5	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	アクティブ データ ファイルの最後の行から行 5 までを逆の順番で読み込みます。
-1 ^a	-1 ^b	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	アクティブ データ ファイルの最後の行を読み込みます。
1	1	順序通りにすべての行を取得	0 または False (順序通りにすべての行を取得)	行 1 を読み込みます。
1	-1 ^b	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	すべての行をランダムな順序で読み込みます。
1	5	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	行 1 から 5 をランダムな順序で読み込みます。
5	-1 ^b	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	アクティブ データ ファイルの行 5 から最後の行までをランダムな順序で読み込みます。
5	1	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	行 5 から 1 をランダムな順序で読み込みます。
-1 ^a	1	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	すべての行をランダムな順序で読み込みます。
-1 ^a	5	ランダムな順序ですべての行を取得	-1 または True (ランダムな順序ですべての行を取得)	アクティブ データ ファイルの行 5 から最後の行までをランダムな順番で読み込みます。
1	-1 ^b	5	5	アクティブ データ ファイルから 5 行をランダムに読み込みます。
1	10	5	5	アクティブ データ ファイルの 1 番目の行から 10 番目の行までのうち、5 行をランダムに読み込みます。
10	-1 ^b	5	5	アクティブ データ ファイルの 10 番目の行から最後の行までのうち、5 行をランダムに読み込みます。
1	5	10	10	範囲外実行時エラーが発生します。

^a ビジュアル テストでは、**開始行** の値に対して **データを含む最後の行から開始** チェック ボックスをオンにすることに相当します。

^b ビジュアル テストでは、**終了行** の値に対して **データを含む最後の行で終了** チェック ボックスをオンにすることに相当します。


アクティブ データ ファイルの行数を超える値を **StartRow**、**EndRow**、または **RandomCount** パラメータに指定すると、範囲外実行時エラーが生成されます。


LoadActiveData() が、別のアプリケーションで開かれているアクティブ データ ファイルを読み込もうとすると、実行時エラーが発生します。

ビジュアル テストでのアクティブ データの使用

ビジュアル テストでアクティブ データを使用するには、使用するすべてのデータ ファイルを作成する必要があります。また、使用する各データ ファイルをアクティブ データ資産に関連付ける必要があります。

Silk Test Workbench データベースの外部からのテスト データの格納と取得は、外部ファイルを使用して行われます。テスト データは、Excel スプレッドシート ファイル (.xls、.xlsx)、テキスト ファイル (.txt)、またはカンマ区切り値ファイル (.csv) に格納できます。テスト データを外部ファイルに格納すると、複数のアクティブ データ資産、および任意のビジュアル テストまたはスクリプトで、テスト データを再利用できるようになります。


 **注:** Microsoft Excel® がインストールされていない場合、すべての .xls ファイルは読み取り専用モードになります。Silk Test Workbench で .xlsx ファイルを使用するには、Microsoft Excel® 2007 以降をインストールしてください。Microsoft Excel® 2007 以降がインストールされていない場合、**2007 Office System** ドライバ：データ接続コンポーネント をインストールして、読み取り専用モードで .xlsx ファイルを使用できます。

 **注:** アクティブ データ ファイルを開くときにエラーが発生した場合は、エラー処理を使用してテスト実行を続行できます。

ビジュアル テストでは、どのアクティブ データ資産がビジュアル テストに使用できるかを指定する必要があります。アクティブ データを使用するビジュアル テストの各ステップのプロパティは、正しいデータを指すように変更する必要があります。これはマッピングと呼ばれます。

アクティブ データ資産のビジュアル テストへの関連付け


アクティブ データ資産をビジュアル テストで使用するには、アクティブ データ資産をビジュアル テストに関連付ける必要があります。1 つのビジュアル テストにアクティブ データ資産をいくつでも関連付けることができます。

 **注:** アクティブ データ資産は、ビジュアル テストの任意のテスト ステップが選択されているときに挿入/作成できます。ただし、プロパティは、<<開始>> ステップの **アクティブ データ** プロパティに表示されます。


1. アクティブ データを使用するビジュアル テストを開きます。
2. 新しいアクティブ データ資産を作成するには、**挿入 > アクティブ データ > 新規作成** を選択します。
3. アクティブ データ資産を参照するには、**挿入 > アクティブ データ > 既存の関連付け** を選択します。

アクティブ データの参照 ダイアログ ボックスが開きます。

- a) 資産リストにすべての資産を表示するか、自分で作成したものだけを表示するかを選択します。
- b) リストから資産を選択します。

 **注:** 列見出し (**名前**、**プロジェクト**、**説明**、**更新日**) をクリックして、その内容を基準にアクティブ データ資産のリストをアルファベット順に並べ替えることができます。

- c) **OK** をクリックします。

 **ヒント:** 関連付けられたアクティブ データ ファイルにデータ行またはデータ列が含まれていない場合、Silk Test Workbench によりメッセージが表示されます。アクティブ データ資産を

ビジュアルテストに関連付けるには **はい** をクリックし、取り消すには **いいえ** をクリックします。

4. 以上のステップを繰り返して、他のアクティブ データ資産のアクティブ データ ファイルを関連付けます。

アクティブ データ資産がビジュアルテストに関連付けられるたびに、項目 [n] という名前の追加プロパティがアクティブ データ プロパティに表示されます。ここで、[n] はビジュアルテストに関連付けられたアクティブ データ資産の合計数のうちの連続番号を示します。選択したアクティブ データ資産の名前は、このプロパティの値として表示されます。**アクティブ データ** プロパティは、**全般** プロパティ カテゴリ内にあります。

5. <<開始>> ステップをクリックして、ビジュアルテストにおけるアクティブ データの関連付けを編集します。

ステップのプロパティが **プロパティ** ペインに表示されます。カテゴリ別またはアルファベット順にプロパティを並び替えます。関連付けを編集するには、**プロパティ** ペインの適切な **項目** プロパティをクリックします。

- **アクティブ データの参照** ダイアログ ボックスを使用して、関連付けられているアクティブ データ資産を置き換えるには、値領域の **アクティブ データの参照** ツールバー ボタンをクリックします。
- ビジュアルテストへのアクティブ データ資産の関連付けを削除するには、**削除** ツールバー ボタンをクリックします。

ビジュアルテストでのアクティブ データに対する繰り返しロジックの作成

アクティブ データに対して繰り返しロジックを作成すると、ビジュアルテストでアクティブ データ ファイルのデータを検索し、ステップ内のリテラル データをアクティブ データ ファイルのデータと置き換えることができます。

ビジュアルテストでアクティブ データの繰り返しロジックを作成する場合、基本的にはアクティブ データテストに使用されるテスト ステップの連続ブロックを定義します。リテラル データをアクティブ データ ファイルのデータと置き換えるよう定義されたブロックのステップは、再生時に繰り返されます。ステップを繰り返す回数は繰り返しロジックで定義され、アクティブ データ ファイルのすべての行または一部の行を対象にすることができます。

ビジュアルテストでアクティブ データの繰り返しロジックを作成しても、ビジュアルテストでアクティブ データが使用されるとは限りません。この処理部分によってテストにステップが設定されますが、ステップ内のデータそのものは、アクティブ データ ファイルのデータにマップする必要があります。

1. ビジュアルテストで、置き換えるデータを含む最初のステップの前にあるステップを選択します。
2. **挿入 > テスト ロジック > 繰り返し** をクリックします。 **テスト ロジック デザイナ** ウィザードが開きます。
3. ウィザードの [ようこそ] 画面を表示する **全般** オプションが **はい** に設定されている場合は、**ようこそ** ページが開きます。 **次へ** をクリックして、**ロジックの種類を選択 (繰り返し)** ページを表示します。



ヒント: ウィザードの [ようこそ] 画面を表示する **全般** オプションが **いいえ** に設定されている場合は、**テスト ロジック デザイナ** ウィザードの最初のページとして **ロジックの種類を選択 (繰り返し)** ページが表示されます。

4. **ロジックの種類を選択 (繰り返し)** ページで、**アクティブ データ ファイル** を使用して **ステップ シーケンスを繰り返す** オプションを選択して、**次へ** をクリックします。 **使用するアクティブ データ資産の定義** ページが表示されます。
5. 使用するデータ ファイルを含むアクティブ データ資産を選択します。
6. 省略可能： **シート名** リストからデータ ファイル中の使用するシートを選択します。デフォルトでは、アクティブ データ資産で指定されているシートが使用されます。
7. **開始行** と **終了行** を選択して、定義したシートのどの行が繰り返し中に使用されるかを定義します。詳細については、「アクティブ データに使用するデータの指定」を参照してください。

- すべての指定したデータ行を順番に繰り返すか、ランダムな順序で繰り返すか、あるいは指定した数だけランダムにデータ行を繰り返すかを定義するには、**ランダム数** から対応するオプションを選択します。

詳細については、「アクティブ データに使用するデータの指定」を参照してください。

- 次へ** をクリックします。



ヒント: 繰り返しの構築 ページを使用して、ビジュアル テストで繰り返すステップのブロックを定義します。**繰り返しの構築** ページを使用してビジュアル テストで繰り返すステップのブロックを定義することは、**アクティブ データ領域の定義** ウィザード ページを使用してテスト スクリプトで繰り返すコードのブロックを定義することと似ています。

繰り返しの構築 ページが開きます。

- アクティブ データ繰り返しロジックに対して繰り返すステップを定義したら、**次へ** をクリックします。**要約** ページが開きます。

- 完了** をクリックします。Silk Test Workbench によって、**テスト ロジック デザイナ** ウィザードの **繰り返しの構築** ページで選択された最初のステップの前、および最後のステップの後ろに、繰り返しロジック ステップが挿入されます。繰り返しロジックを開始する、挿入されたステップのステップ テキストは次のようになります: アクティブ データ '[資産名]' を使用して繰り返します

繰り返しロジックを終了する、挿入されたステップのステップ テキストは次のようになります: 繰り返しの終了

アクティブ データ ファイルのデータのビジュアル テスト データへのマップ

ビジュアル テストでアクティブ データ ファイル内のデータを使用するには、該当するテスト ステップのデータをアクティブ データ ファイル内のデータ列にマップする必要があります。これには、まず目的のアクティブ データ ファイルを含むアクティブ データ資産をビジュアル テストに関連付けます。次に、アクティブ データ ファイルのデータを使用する各ステップを、該当するデータを含むアクティブ データ ファイルの列にマップする必要があります。この手順を使用して、テスト ステップ内のデータをアクティブ データ ファイルのデータにマップします。



注: マップされたデータに対してアクティブ データ ファイルのデータを使用した繰り返しロジックが作成されていない場合は、データをアクティブ データ ファイルのデータ列にマップしても、再生中に列の最初のレコードしか使用されません。

- ビジュアル テストで、ビジュアル テストに関連付けられたアクティブ データ ファイルのデータと置き換えるデータを含む最初のステップを選択します。

ステップのプロパティが **プロパティ** ペインに表示されます。プロパティは、カテゴリ別またはアルファベット順に並び替えることができます。アクティブ データ ファイルのデータを使用した繰り返しロジックがビジュアル テストにすでに作成されている場合は、マップされるデータを含むステップが「アクティブ データを使用して繰り返します」ステップと「繰り返しの終了」ステップの間にネストされている必要があります。

- ステップの **プロパティ** ペインで、置き換えるデータを含むプロパティの値領域をクリックします。
- 値領域の **選択** ボタンをクリックして、**アクティブ データ** をクリックします。**アクティブ データ列の選択** ダイアログ ボックスが開きます。**アクティブ データ資産** テキスト ボックスに、ビジュアル テストと関連付けられたアクティブ データ資産が表示されます。アルファベット順の最初の資産の名前がボックスに表示されます。
- アクティブ データ資産** テキスト ボックスで、マップするデータを含むアクティブ データ資産の名前を選択します。

選択したアクティブ データ資産のアクティブ データ ファイルの列名が、**列** リストに表示されます。

- 選択したアクティブ データ資産のアクティブ データ ファイルが現在元のアプリケーション (テキスト エディタまたは Microsoft Excel) で開かれていると、列名が **列** リストに表示されません。この場合は、元のアプリケーションを閉じ、マップするデータを含むアクティブ データ資産を再度選択します。

- アクティブ データ資産に対して **最初の行を見出しとして使用** チェック ボックスがオンになっている場合は、データ ファイルの最初の行のデータが列見出しとみなされます。各列の最初の行のデータが、**列** リストに表示される列名となります。
- アクティブ データ資産に対して **最初の行を見出しとして使用** チェック ボックスがオフになっている場合は、データ ファイルの最初の行のデータがデータとみなされます。この場合、Silk Test Workbench は **列** リスト内の名前に対して英字を順番に使用します (「A」、「B」、「C」)。



ヒント: アクティブ データ ファイルを作成する場合は、可能なかぎり、名前付きの見出しを使用します。**アクティブ データ列の選択** ダイアログ ボックスの **列** リストにわかりやすい列名が表示されると、アクティブ データ ファイルの正しい列にデータをマップしやすくなります。

5. **列** リストで、選択したプロパティで使用される実際のデータと置き換えるデータを含む列名を選択します。
6. **OK** をクリックして **アクティブ データ列の選択** ダイアログ ボックスを閉じ、データをマップします。Silk Test Workbench は、アクティブ データ ファイルの選択されたデータにマップする式を使用して実際のデータを置き換えます。式の構文は以下のとおりです: [[アクティブ データ資産名.データ型]("列名")]。ここで、[アクティブ データ資産名.データ型] は、選択されたアクティブ データ資産の名前および選択された列内のデータのデータ型です。ビジュアル テストの場合、アクティブ データ ファイルで使用されるデータは Silk Test Workbench により自動的に解釈されます。データに数字のみが含まれる場合、データ型は Long になります。データに 1 文字以上の英字が含まれる場合、データ型はテキストになります。

("列名") は、プロパティのデータがマップされる列の名前です。



ヒント: リテラル、変数、式、またはアクティブ データを選択して設定した値は、元の値が存在するときにビジュアル テストを保存していた場合にはリセットすることができます。プロパティの値をリセットするには、値領域で **選択** ボタンをクリックして、元のリテラル、変数、式の結果、またはアクティブ データのいずれかのタイプを割り当てます。データ型の元の値がプロパティ値として表示されます。

7. ビジュアル テストに関連付けられたアクティブ データ ファイルのデータと置き換えるデータを含む他のすべてのテスト ステップについて、以上のステップを繰り返します。
マップされたデータに対してアクティブ データ ファイルのデータを使用した繰り返しロジックが作成されていない場合は、データをアクティブ データ ファイルのデータ列にマップしても、再生中に列の最初のレコードしか使用されません。



注: 同じビジュアル テストで複数のアクティブ データ資産を使用することができますが、1 つの繰り返し文内で使用できるアクティブ データ資産は 1 つだけです。

アクティブ データによるビジュアル テストのステップ テキストの更新

ステップ内のデータをアクティブ データ ファイルのデータにマップすると、Silk Test Workbench によってステップ テキストが更新されます。ステップでは、**テスト ステップ** ペインの **ステップ** 列のリテラル データが、アクティブ データ資産、置き換えられるデータ型、および置き換えられるデータを含むアクティブ データ ファイルの列を示すテキストと置き換えられます。

以下の例に、Silk Test でステップ テキストのリテラル データがアクティブ データ ファイル内のマップされたデータとどのように置き換えられるかを示します。

例 1 : テキスト入力

以下の例は、テスト アプリケーションの TextField コントロールにリテラル値 Smith を入力する自動ステップを示しています。リテラル値は、ステップの **テキスト** プロパティに設定されています。

'Smith' を入力します

リテラル データが **contacts** という名前のアクティブ データ資産にある **Last Name** という名前の列のデータで置き換えられると、ステップ テキストは以下のように変わります。

'[[contacts].Text("Last Name")]' を入力します

この場合、置き換えられたデータには英字が含まれるため、Silk Test Workbench は置き換えられた値をテキストとして解釈し、ステップで指定されるデータ型が Text になります。

例 2 : 数値入力

以下の例は、ステップの Left および Top プロパティの座標値に基づいて画面上の特定の位置にウィンドウを移動するステップを示しています。

(235, 427) に移動します

Left および Top プロパティの値が **coordinates** という名前のアクティブ データ資産に関連付けられたファイル内の **Left Coordinate** および **Top Coordinate** という名前の列のデータに置き換えられると、ステップ テキストは以下のように変わります。

([[>coordinates].Long("Left Coordinate")], [[coordinates].Long("Top Coordinate")]) に移動します

この場合、置き換えられたデータには数値のみが含まれるため、Silk Test Workbench はアクティブ データ ファイルの値を数値として解釈し、ステップで指定されるデータ型が Long になります。

ビジュアル テストでのアクティブ データ ファイルのデータの更新

ビジュアル テストに関連付けられているアクティブ データ ファイルのデータは、ビジュアル テスト内で更新することができます。ビジュアル テストでアクティブ データを更新するには、式を使用します。**式デザイナー**を使用すると、アクティブ データ ファイルの列のデータを入力として使用でき、データの値を変更する式を作成できます。更新されたデータは、同じアクティブ データ ファイルの同じ列に保存できます。

アクティブ データ テストの繰り返しロジック内で使用すると、アクティブ データ ファイルの列にある各行のデータを式を介して更新し、列内の元の行に格納することができます。これには、式内で列を使用し、式でデータを更新（列内の各行のデータを更新）してから、式の内容をアクティブ データ ファイルの同じ列に戻します。また、関連付けられている他のビジュアル テストの別の列に式の内容を格納したり、更新されたデータをローカル変数に格納したりすることもできます。ローカル変数は、テスト ロジックや結果コメントで使用でき、グローバル変数または他のビジュアル テストやスクリプトに渡すことができます。

式を使用してビジュアル テストで更新されたアクティブ データは、ビジュアル テストの再生が正常に完了した時点で自動保存されます。ただし、ビジュアル テストを変更して、更新されたデータの即時保存、再生中の任意の時点での保存、自動保存の回避を行うこともできます。

ビジュアル テストでの式を使用したアクティブ データの更新

ビジュアル テストに関連付けられているアクティブ データ ファイルのデータは、ビジュアル テスト内で更新することができます。これを行うには、**式デザイナー**を使用して、データの指定、更新を行います。更新されたデータを変数に保存してその変数をビジュアル テストで使用したり、元のアクティブ データ ファイルに保存したりできます。

1. ビジュアル テストで、作成する式で使用する任意のローカル変数を作成します。
2. ビジュアル テストで、作成する式で使用するデータ列を含むアクティブ データ資産を関連付けます。
3. 式ステップを作成する場所の直前のステップを選択します。




ヒント: アクティブ データ ファイルのデータを使用する式は、アクティブ データを使用する繰り返しロジックの外側を含め、ビジュアル テスト内の任意の場所に配置できます。ただし、式で更新されたデータをアクティブ データ列の元の行にそれぞれ保存する場合は、繰り返しロジック内でデータを更新するステップの直後に配置する必要があります。Expression ステップを繰り返しロジックのあとに配置すると、アクティブ データ ファイルで最後にアクセスされた行のデータのみが更新されます。

4. **挿入 > 式** を選択します。**式デザイナー** ダイアログ ボックスが開きます。
5. **式デザイナー** ダイアログ ボックスを使用して、アクティブ データ ファイルの列のデータを更新する式を作成します。
列内のアクティブ データを更新し、内容を同じ列に戻すには、**式デザイナー** ダイアログ ボックスの **式エディタ** を使用して値を更新してから、式の内容を同じ列に戻します。

たとえば、**式エディタ** でデータ ファイルの「Gender」という列（列は string データ型で、値は「M」または「Fem」）を入力として使用する場合、**式エディタ** で、データに「ale」を追加して「Male」または「Female」にします。その後、データはダイアログ ボックスの **出力** セクションに示された同じ列に戻されます。

6. 式を作成してテストしたあと、**式デザイナー** ダイアログ ボックスで **OK** をクリックします。Silk Test Workbench によって、選択したステップのあとに 式：ステップが挿入されます。

 **ヒント:** 作成したあとで式を編集するには、式：ステップを選択し、ステップの 式プロパティを選択して、式デザイナー をクリックし、**式デザイナー** ダイアログ ボックスを使用して式を編集します。


式で更新されたアクティブ データは、再生が正常に完了した時点で元のアクティブ データ ファイルに自動保存されます。ただし、元のアクティブ データ ファイルに更新内容を即時保存するステップを作成したり、保存を取り消して更新されたデータの保存を回避するステップを作成したりすることもできます。

ビジュアル テスト内で更新されたアクティブ データの即時保存

ビジュアル テストの再生中に式内で更新されるアクティブ データは、再生が正常に完了した時点で元のアクティブ データ ファイルに自動保存されます。アクティブ データ ファイルのさまざまなセルのデータが変更されている場合、他のセルが指定されていないかぎり、Silk Test Workbench は更新されたデータを同じセルに保存します。


再生が正常に完了するまでデータが元のアクティブ データ ファイルに自動保存されるのを待つ代わりに、更新された特定のデータを直ちに保存するステップを作成することができます。このステップでは、ビジュアル テストでこのステップを実行するまでに更新された選択済みのアクティブ データのみが保存されます。ステップの実行後に更新されたアクティブ データに関しては、自動保存を取り消さないかぎり、再生が正常に完了した時点で保存されます。

1. ビジュアル テストで、作成する式で使用する任意のローカル変数を作成します。
2. ビジュアル テストで、作成する式で使用するデータ列を含むアクティブ データ資産を関連付けます。
3. アクティブ データ ファイルのデータを更新し、元のセルに更新内容を格納する式を作成します。
4. ビジュアル テスト内で、データを更新するステップのうち、直ちに保存を行う最後の 式:ステップを選択します。
5. **挿入 > アクティブ データ > 直ちに保存** を選択します。Silk Test Workbench によって、選択したステップの下に「アクティブ データ ファイル " に値を直ちに保存します」ステップが挿入されます。
6. 作成した「アクティブ データ ファイル " に値を直ちに保存します」ステップを選択します。ステップのプロパティが **プロパティ** ペインに表示されます。
7. アクティブ データ カテゴリ プロパティを設定して、更新された値を直ちに保存するデータ ファイルを含むアクティブ データ資産を指定します。

 **ヒント:** 必要に応じて、「アクティブ データ ファイル " に値を直ちに保存します」ステップの **アクティブ データ** プロパティを使用して、保存オプションや保存するアクティブ データを変更します。

ビジュアル テスト内で更新されたアクティブ データの自動保存の回避

ビジュアル テストの再生中に式内で更新されるアクティブ データは、再生が正常に完了した時点で元のアクティブ データ ファイルに自動保存されます。ただし、更新されたアクティブ データの自動保存を取り消すステップを作成することができます。

 **注:** このステップでは、このステップを実行するまでに更新されたアクティブ データの保存のみが回避されます。このステップの実行後に更新されたアクティブ データに関しては、更新後に自動保存を取り消さないかぎり元のアクティブ データ ファイルに保存されます。

1. ビジュアル テストで、アクティブ データ ファイルのデータを更新し、元のセルに更新内容を格納する式を作成します。
2. ビジュアル テスト内で、データを更新するステップのうち、自動的に保存を行わない最後の 式:ステップを選択します。

3. **挿入 > アクティブ データ > 保存のキャンセル** を選択します。Silk Test Workbench によって、選択したステップの下に「アクティブ データ ファイル " に値を保存しません」ステップが挿入されます。
4. 作成した「アクティブ データ ファイル " に値を保存しません」ステップを選択します。ステップのプロパティが **プロパティ** ペインに表示されます。
5. ステップの **アクティブ データ** カテゴリプロパティを設定して、更新された値を直ちに保存しないデータ ファイルを含むアクティブ データ資産を指定します。

ビジュアル テストでアクティブ データに使用するシートを指定する

アクティブ データ資産のソースとして Excel スプレッドシートを定義し、このアクティブ データ資産をビジュアル テストでのアクティブ データ資産テストに使用した場合、ビジュアル テストで実行中に使用するシートを直接指定できます。

1. ビジュアル テストを開きます。
2. **アクティブ データ <資産名> を使用して繰り返します** ステップを選択します。
3. ステップの **プロパティ** ペインを開きます。
4. **条件** 領域の **シート名** フィールドにシートの名前を入力します。
ビジュアル テストの実行時に、ここで指定したシートで、アクティブ データ資産で指定したシートが上書きされます。このフィールドを空のままにすると、資産で指定したシートが使用されます。
5. ビジュアル テストを保存します。

指定したシートが Excel スプレッドシート内に存在しない場合、ビジュアル テストの実行時に Silk Test Workbench はエラーをスローします。この場合、Excel スプレッドシートを確認して存在するシートを指定してください。

アクティブ データ資産の行数と列数の取得

アクティブ データ資産の行数や列数を取得するには、資産に関連付けられたビジュアル テストに 行数の取得 または 列数の取得 ステップをそれぞれ追加します。


1. これらのステップは、次のようにしてビジュアル テストに追加できます。
 - Silk Test Workbench メニューから、**挿入 > アクティブ データ > 行数の取得** または **挿入 > アクティブ データ > 列数の取得** を選択します。
 - ビジュアル テストの **テスト ステップ** ペインで、**操作 > 挿入 > アクティブ データ > 行数の取得** または **操作 > 挿入 > アクティブ データ > 列数の取得** を選択します。
 - ビジュアル テストの **テスト ステップ** ペインで右クリックして、**挿入 > アクティブ データ > 行数の取得** または **挿入 > アクティブ データ > 列数の取得** を選択します。
2. これらのステップの **プロパティ** ウィンドウで、次の操作を実行します。
 - a) 使用する関連付けられたアクティブ データ資産を選択します。
 - b) 行数と列数のどちらを取得するかを選択します。
 - c) 結果を保存する変数を選択します。
結果は常に `st_LastReturnValue` 予約変数に保存されます。このため、ローカル変数の選択は必須ではありません。また、テスト結果の **詳細** ペインでは、テスト ステップの **結果の詳細** 列にも結果が保存されます。


行数の取得 または 列数の取得 ステップがアクティブ データを使用して繰り返します ループ内で使用される場合、行数はアクティブ データを使用して繰り返します ステップで設定された開始行と終了行に基づきます。たとえば、アクティブ データ ファイルに 10 行保存されている場合に、アクティブ データを使用して繰り返します ループで 2 行目から 6 行目だけを使用するように設定していると、行数の取得 ステップは 5 を返します。

ステップをアクティブ データを使用して繰り返します ループの外で使用した場合は、アクティブ データ ファイルの実際の行数と列数が返されます。

スクリプトでのアクティブ データの使用

アクティブ データを使用するスクリプトの作成処理は、記録機能から独立しています。Silk Test Workbench データベースの外部からのテスト データの格納と取得は、外部ファイルを使用して行われます。テスト データは、Excel スプレッドシート ファイル (.xls、.xlsx)、テキスト ファイル (.txt)、またはカンマ区切り値ファイル (.csv) に格納できます。テスト データを外部ファイルに格納すると、複数のアクティブ データ資産、および任意のビジュアル テストまたはスクリプトで、テスト データを再利用できるようになります。

 **注:** Microsoft Excel® がインストールされていない場合、すべての .xls ファイルは読み取り専用モードになります。Silk Test Workbench で .xlsx ファイルを使用するには、Microsoft Excel® 2007 以降をインストールしてください。Microsoft Excel® 2007 以降がインストールされていない場合、2007 Office System ドライバ：データ接続コンポーネント をインストールして、読み取り専用モードで .xlsx ファイルを使用できます。

 **注:** アクティブ データ ファイルを開くときにエラーが発生した場合は、エラー処理を使用してテスト実行を続行できます。


アクティブ データ テスト資産の作成

新しいアクティブ データ資産を作成するには、以下を実行します。

1. **表示 > アセットブラウザ** を選択します。
2. **資産の種類** で、**アクティブ データ** を右クリックして、**アクティブ データの新規作成** を選択します。**アクティブ データ** ウィンドウが開きます。

3. **名前** テキスト ボックスに、資産のわかりやすい名前を入力します。

アクティブ データ資産名は、Silk Test Workbench 資産の命名規則に従う必要があります。

 **ヒント:** アクティブ データ資産に、その資産に使用されるアクティブ データ ファイルと同じ名前を付けると、テストで使用する資産を特定しやすくなります。


4. 資産が使用するデータを含んだ新しいアクティブ データ ファイルを作成するか、既存のアクティブ データ ファイルを選択できます。

- 新しいアクティブ データ ファイルをファイル システムに作成するには、**外部ファイル** を選択してから **新規作成** をクリックします。
- 既存のアクティブ データ ファイルのデータを使用するには、**外部ファイル** を選択してから **参照** をクリックし、ファイルを選択します。アクセスできるようにするため、ファイルを閉じてください。
- 新しいアクティブ データ資産を Silk Test Workbench データベースに作成するには、**内部データ** を選択してから **新規作成** をクリックします。
- 既存のアクティブ データ ファイルを Silk Test Workbench データベースに格納して、アクティブ データ資産を使って内部ファイルからデータを使用するには、**内部データ** を選択してから **アップロード** をクリックし、ファイルを選択します。
- 既存のアクティブ データ ファイルを Silk Test Workbench データベースからファイル システムにコピーするには、**内部データ** を選択してから **ダウンロード** をクリックし、ファイルを選択します。保存したファイルは、Excel などのアプリケーションを使って編集できます。

ファイル テキスト ボックスには、アクティブ データ ファイルの名前が表示されます。

5. **省略可能 : 説明** テキスト ボックスに、資産の説明を入力します。

説明を使用して、データ ファイルとして使用されるファイルの種類、資産を作成するアクティブ データ テストによって実行されるテストの種類、またはアクティブ データ テストでテストされるアプリケーションを示すことができます。**アセットブラウザ** では、説明が入力されているすべてのアクティブ データ資産に説明が表示されます。

 **注:** 空のデータ ファイルが必要な場合は、**保存** および **閉じる** をクリックしてすべての変更内容を保存し、**アクティブ データ** 設定ウィンドウを閉じます。

6. **アクティブ データ** ウィンドウで、**オプション** タブに切り替えます。
7. アクティブ データ ファイルに対して読み取りオプションを設定します。
8. **アクティブ データ** ウィンドウで、**詳細** タブに切り替えます。
9. アクティブ データ ファイル内のデータを編集します。
- 10 **保存** または **データの保存** をクリックしてすべての変更内容を保存するか、または **保存して閉じる** をクリックしてすべての変更内容を保存し、**アクティブ データ** ウィンドウを閉じます。
 アクティブ データ資産のデータ ファイルが外部の場合、つまりファイル システムに格納されており、Silk Test Workbench データベースではない場合、**データの保存** をクリックすると、データ ファイルだけが保存されます。この場合、アクティブ データ資産は保存されません。

新しいアクティブ データ資産が **アセットブラウザ** に表示されます。

スクリプトへのアクティブ データの組み込み

開始する前に、使用するデータを含むアクティブ データ資産、またはデータを書き込むアクティブ データ資産を作成します。

アクティブ データ資産から使用するスクリプトにデータをマップします。さらに、アクティブ データ資産に行または列を追加して、スクリプトからアクティブ データ資産にデータを追加できます。

1. テストする操作を記録するか、またはテストする操作についてのスクリプトを手動で作成します。
 詳細については、「スクリプトの記録」を参照してください。
 たとえば、Web アプリケーションで姓、名、および電話番号を記録するスクリプトは以下のようになります。

```
Public Sub Main()
    With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")
        .DomTextField("@id='txtFirstName']").SetText("Pat")
        .DomTextField("@id='txtLastName']").SetText("Smith")
        .DomTextField("@id='txtPhoneNumber']").SetText("555-121-3434")
        .DomButton("@id='btnAdd']").Select()
    End With
End Sub
```

2. LoadActiveData コマンドを使用して、アクティブ データ資産名を指定します。
 必要に応じて、アクティブ データ ファイル内の開始行と終了行、使用するランダム行数、およびデータが読み取り専用であるかどうかを指定できます。

```
Workbench.LoadActiveData("activeDataName", [start row,
    end row, randomcount, readonly])
```
3. 記録したテキストをアクティブ データ資産のテキストで置き換えるには、アクティブ データ ファイル内の対応する各行名に対して宣言を作成し、宣言をスクリプト内の適切なコードにマップします。
 たとえば、PhoneBookData アクティブ データ資産で利用可能なすべてのデータを使用するようにステップ 1 のスクリプトを変更するには、スクリプトを以下のように変更します。

```
Public Sub Main()
    AddAllToPhoneBook() 'Reads all records in order
End Sub

Public Sub AddAllToPhoneBook()
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData" )
    Dim row As ActiveDataRow

    With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")

        For Each row In data

            Dim FirstName As String = row.GetString("fname")
            Dim LastName As String = row.GetString("lname")
            Dim PhoneNumber As String = row.GetString("phonenumber")
        End For
    End With
End Sub
```

```

.DomTextField("@id='txtFirstName").SetText(FirstName)
.DomTextField("@id='txtLastName").SetText(LastName)
.DomTextField("@id='txtPhoneNumber").SetText(PhoneNumber)

.DomButton("@id='btnAdd").Select()
Next
End With
End Sub

```

スクリプトを再生すると、アクティブ データ資産のすべての宣言されているデータがスクリプトに組み込まれます。

4. 既存のアクティブ データ資産に列を追加するには、アクティブ データ資産を宣言して、アクティブ データ資産名と列名を指定します。

たとえば、PhoneBookData 資産に「Occupation」という新しい列を追加するには、以下のように入力します。

```

Public Sub Main()

    AddColumnToDataFile( "occupation" )
End Sub

Public Sub AddColumnToDataFile( name As String )
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData" )
    data.AddColumn( "Occupation", name )
    data.Save()
End Sub
End Module

```

スクリプトを再生すると、アクティブ データ資産によって指定されたファイルに新しい列が追加されます。

5. 既存のアクティブ データ資産に行を追加するには、アクティブ データ資産を宣言して、アクティブ データ資産名と行名を指定します。

たとえば、特定の ID 番号の新しいデータ行を追加するには、以下のように入力します。

```

Public Sub Main()

    InsertNewNumberToDataFileWithOccupation( 108, "Jay", "Jones", "5551219", _
        "QA Analyst" )
End Sub

Public Sub InsertNewNumberToDataFileWithOccupation( id As Integer, _
    firstName As String, lastName As String, phoneNumber As String, _
    occupation As String )
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData" )
    Dim row As ActiveDataRow = data.AddRow()

    row.SetLong( "id", id )
    row.SetString( "fname", firstName )
    row.SetString( "lname", lastName )
    row.SetString( "number", phoneNumber )
    row.SetString( "occupation", occupation )

    data.Save()

End Sub
End Module

```

スクリプトを再生すると、アクティブ データ資産によって指定されたファイルに新しい行が追加されます。

6. 特定のセルに変数の値を追加するには、スクリプトに次のコード サンプルを追加します。

```
Dim row As ActiveRecord = data.Item(1)
row.SetString("Columnname", Variable)
```

VB .NET スクリプトでアクティブ データに使用するシートを指定する

アクティブ データ資産のソースとして Excel スプレッドシートを定義し、このアクティブ データ資産を VB .NET スクリプトでのアクティブ データ資産テストに使用した場合、LoadActiveData メソッドの sheetName パラメータを使用して、実行中に使用するシートを指定できます。詳細については、「LoadActiveData メソッド」を参照してください。

指定したシートが Excel スプレッドシート内に存在しない場合、VB .NET スクリプトの実行時に Silk Test Workbench はエラーをスローします。この場合、Excel スプレッドシートを確認して存在するシートを指定してください。

ビジュアル テストの拡張


このセクションでは、ビジュアル テストを変更して、より高度な機能を実行する方法を示します。

ビジュアル テスト ステップで実行される操作の変更

操作の選択 ダイアログ ボックスを使用して、ビジュアル テストのステップのコントロールで実行される操作を変更できます。

リストには以下の列が含まれています。

操作をクリックして選択し、**OK** をクリックすると、テスト ステップでコントロールに対してその操作が実行されるようになります。

 **注:** コントロールに対して、選択された操作を特定の方法で実行するために、ほとんどの操作で追加パラメータが使用されます。操作のパラメータは、**パラメータ** カテゴリにプロパティとして表示され、**プロパティ** ウィンドウで設定できます。

1. 変更する操作が含まれているビジュアル テストを開きます。
2. **テスト ステップ** ペインで、変更する操作を含むステップを選択します。
3. **プロパティ** ペインを開きます。
4. ペインの **操作** 列で、**メソッド** をクリックします。
5. **操作の選択** をクリックします。**操作の選択** ダイアログ ボックスが開き、現在のテスト ステップがやり取りする UI コントロールで利用可能なすべての操作が一覧表示されます。リストの各列のテキストを確認できるようダイアログ ボックスのサイズを変更できます。
6. UI コントロールで実行したい操作を選択します。
7. **OK** をクリックします。

ビジュアル テストへのコメントの挿入

コメントによって、ビジュアル テストにステップについての補足情報を定義できます。既存のビジュアル テストにステップとしてコメントを挿入できます。

1. ビジュアル テストの **テスト ステップ** ペインで、コメントを後ろに挿入するステップを選択します。
2. **挿入 > コメント** を選択します。**テスト ステップ** ペインに コメント ステップが追加され、このステップのプロパティが **プロパティ** ペインに表示されます。
3. **プロパティ** ペインで、**コメント** フィールドにコメントのテキストを入力します。

フラグ

フラグを使用して、ビジュアル テスト内または結果内の問題について自分自身や別のチーム メンバに注意を促すことができます。問題にフラグを付けると、挿入したフラグの種類に応じて、**フラグ** のアイコンまたは **割り当てられたフラグ** のアイコンが追加されます。

フラグには以下の情報が含まれます。

- フラグを作成または更新したユーザーの名前
- フラグの作成日または更新日
- フラグの説明（オプション）
- フラグが割り当てられたユーザーの名前（オプション）
- ビジュアル テストまたは結果が保存されているプロジェクト（オプション）

フラグ のアイコンは、**テスト ステップ** ペインの **フラグ** 列に表示されます。フラグにポインタを合わせると、ツール ヒントとしてフラグの内容が表示されます。

以下のいずれかの方法を使用して、フラグを挿入します。

- ビジュアル テストまたは結果の **テスト ステップ** ペインでステップを選択し、フラグを挿入します。

必要に応じて、フラグに説明を付けたり、別のユーザーにフラグを割り当てたりできます。割り当てられたフラグは、ビジュアル テストまたは結果の選択したステップ、およびフラグを割り当てたユーザーの **開始画面** の **フラグ** ペインに表示されます。**開始画面** では、自身に割り当てられているすべてのフラグをすばやく参照したり、フラグをダブルクリックして対象のビジュアル テストまたは結果を開いたりできます。

- 再生時に失敗した時にビジュアル テストの結果にフラグを送信する検証ロジックをビジュアル テストに作成します。

結果では、検証ロジックを含むステップにフラグが挿入されます。このステップは、結果の **失敗** タブ、**フラグ** タブ、および **詳細** タブの **テスト ステップ** ペインに表示されます。必要に応じて、ウィザードでフラグの説明を作成できます。デフォルトでは、このフラグは自動的に現在のユーザーに割り当てられます。フラグを別のユーザーに割り当てる場合は、再生後に結果から再割り当てする必要があります。



注: フラグはビジュアル テストでのみサポートされており、Visual Basic .NET スクリプトではサポートされていません。

[テスト ステップ] ペインへのフラグの挿入

ビジュアル テストまたは結果での問題を忘れないように、フラグを挿入できます。

1. フラグを挿入するビジュアル テストまたは結果を開きます。
2. ビジュアル テストまたは結果の **合格** タブ、**失敗** タブ、**フラグ** タブ、または **詳細** タブの **テスト ステップ** ペインで、フラグを挿入するステップを選択します。
3. **操作 > フラグ > 挿入** をクリックします。
また、フラグを挿入するステップの **フラグ** 列のセルをダブルクリックして、**フラグ** ダイアログ ボックスにアクセスすることもできます。
フラグ ダイアログ ボックスが開きます。
4. 省略可能： **説明** テキスト ボックスにフラグの説明を入力します。
この説明は、ポインタを **フラグ** アイコンまたは **割り当てられたフラグ** アイコンに合わせると、ツール ヒントとして表示されます。
5. 省略可能： **開始画面に表示してユーザーに割り当てる** チェック ボックスをオンにし、ユーザーを選択して、別のユーザーにフラグを割り当てます。フラグは、割り当て先のユーザーの **開始画面** に表示されます。
6. **OK** をクリックします。**フラグ** ダイアログ ボックスで選択したオプションに応じて、**フラグ** のアイコンまたは **割り当てられたフラグ** のアイコンが、選択したステップの **フラグ** 列に表示されます。結果の

フラグの場合、OK をクリックするとすぐにフラグが保存されます。ビジュアル テストのフラグの場合、フラグを保存するにはビジュアル テストを保存する必要があります。

結果へのフラグの挿入

再生時に検証ロジックに失敗すると、Silk Test Workbench によって、ビジュアル テストの結果にフラグが送信されます。

1. ビジュアル テストを作成または記録します。
2. **テスト ロジック デザイナ** ウィザードを使用して、検証ロジックを挿入します。
3. **検証に失敗した場合に、結果に 'フラグ' を作成する** チェック ボックスをオンにし、フラグの説明を指定します。
4. ビジュアル テストを再生します。再生時に検証ロジックに失敗すると、Silk Test Workbench によって、ビジュアル テストの結果にフラグが送信されます。結果では、検証ロジックを含むステップにフラグが挿入されます。このステップは、**結果** ウィンドウの **失敗** タブ、**フラグ** タブ、および **詳細** タブに表示されます。

フラグの表示


フラグの内容を表示します。

1. フラグを表示するビジュアル テストまたは結果を開きます。
ビジュアル テストまたは結果の **合格** タブ、**失敗** タブ、**フラグ** タブ、または **詳細** タブの **テスト ステップ** ペインで、フラグが付けられたステップには **フラグ** 列に **フラグ** のアイコンまたは **割り当てられたフラグ** のアイコンが表示されます。
2. **フラグ** のアイコンまたは **割り当てられたフラグ** のアイコンにポインタを合わせます。ツール ヒントにフラグの内容が表示されます。

開始画面からのフラグの表示

開始画面 からフラグの内容を表示します。

1. **開始画面** を開きます。デフォルトでは、フラグは作成された日付と表示される資産でグループ化され、**フラグ** ペインに表示されます。

 **注:** **フラグ** ペインの表示を変更するには、**開始画面のフラグ** オプションを変更します。このオプションは、**開始画面** の **ヘルプ** ペインの下にある **カスタマイズ** ボタンをクリックして、**開始画面** からアクセスできます。

2. 表示するフラグに移動して、**割り当てられたフラグ** のアイコンにポインタを合わせます。ツール ヒントにフラグの内容が表示されます。
3. フラグが挿入されているビジュアル テストまたは結果を開くには、**割り当てられたフラグ** のアイコンをダブルクリックします。

フラグの編集

必要に応じて、フラグの内容を変更します。

1. 編集するフラグを含むビジュアル テストまたは結果を開きます。
2. ビジュアル テストまたは結果の **合格** タブ、**失敗** タブ、**フラグ** タブ、または **詳細** タブの **テスト ステップ** ペインで、編集するフラグを含むステップを選択します。
3. **操作** > **フラグ** > **編集** をクリックします。**フラグ** ペインが開きます。また、**フラグ** のアイコンまたは **割り当てられたフラグ** のアイコンをダブルクリックして **フラグ** ダイアログ ボックスを開くこともできます。
4. 省略可能：説明 テキスト ボックスにフラグの説明を入力します。
この説明は、ポインタを **フラグ** アイコンまたは **割り当てられたフラグ** アイコンに合わせると、ツール ヒントとして表示されます。

5. 省略可能 : **開始画面に表示してユーザーに割り当てる** チェック ボックスをオンにし、ユーザーを選択して、別のユーザーにフラグを割り当てます。フラグは、割り当て先のユーザーの **開始画面** に表示されます。
6. **OK** をクリックします。結果のフラグの場合、**OK** をクリックするとすぐにフラグが保存されます。ビジュアル テストのフラグの場合、フラグを保存するにはビジュアル テストを保存する必要があります。

フラグの削除

フラグが不要になったら、削除します。

1. 削除するフラグを含むビジュアル テストまたは結果を開きます。
2. ビジュアル テストまたは結果の **合格** タブ、**失敗** タブ、**フラグ** タブ、または **詳細** タブの **テスト ステップ** ペインで、削除するフラグを含むステップを選択します。
3. **操作 > フラグ > 削除** をクリックします。フラグの削除を確認するメッセージが表示されます。
4. **はい** をクリックします。結果のフラグの場合、**OK** をクリックするとすぐにフラグが削除されます。ビジュアル テストのフラグの場合、フラグを永久的に削除するにはビジュアル テストを保存する必要があります。

テスト ロジックの追加

記録されたビジュアル テストの基本的な機能を拡張するテスト ロジックを含めることで、ビジュアル テストの機能と柔軟性を強化できます。テスト ロジックを挿入すると、ビジュアル テストの実行フローを動的に変更できます。たとえば、ユーザー定義の条件を評価するテスト ロジックをビジュアル テストに挿入して、その条件の結果に基づいて以下のいずれかの操作を実行できます。

- ビジュアル テストの一連のテスト ステップを実行したり、繰り返したりする
- ビジュアル テストからビジュアル テストを実行する
- 合格/失敗メッセージ、およびテスト ステップの再生ステータスに関するフラグをビジュアル テストの結果に送信する

これらの操作を実行するために、Silk Test Workbench にはさまざまなタイプのテスト ロジックが用意されています。これらのロジックタイプは以下のカテゴリに分けられます。

- **判断ロジック** : ユーザー定義の条件を評価して、その条件の戻り値に応じて一連のテスト ステップを実行するロジックです。
- **繰り返しロジック** : 指定した回数または時間、あるいはユーザー定義の条件が true である間、一連のテスト ステップを繰り返します。
- **検証ロジック** : ユーザー定義の条件を評価して、合格/失敗メッセージとフラグをビジュアル テストの結果に送信します。
- **エラー処理ロジック** : 再生中にエラーが発生した場合、ユーザー定義の操作を実行します。

Silk Test Workbench には、テスト ロジックをビジュアル テストに挿入するために、以下のツールが用意されています。

- **テストロジックデザイナーウィザード**。
- **ロジックツールボックス**。
- 判断テスト ステップ *If*、*Else If*、*Else*。

判断ロジック

判断ロジックでは、ユーザー定義の条件を評価して、その条件の戻り値に応じて一連のテスト ステップを実行します。判断ロジックで使用される条件は、2 つの値を比較し、その結果に基づいて値を返します。詳細については、「条件デザイナー」を参照してください。条件内の値には、テスト アプリケーション内のコントロールのプロパティの値、ビジュアル テストで使用される変数の値、またはリテラル値のいずれかを使用できます。詳細については、「テストロジックデザイナー : ロジックの種類を選択 (判断)」を参照してください。

判断ロジックは、以下の要素で構成されます。

```
If <条件=TRUE>
  __<テスト ステップを順番に実行>
End If
```

たとえば、以下の判断ロジックの条件では、変数 *propertyVar* とリテラル値「4」を比較しています。変数 *propertyVar* がリテラル値「4」より大きい場合は、ステップ 5 から 9 が実行されます。

```
If propertyVar より大きい 4
  __ステップ 5 からステップ 9 まで実行
End If
```

また、Else If と Else を使用して、任意のレベル数にネストした複雑な判断ロジックを作成することもできます。

```
If <条件=TRUE>
  __<テスト ステップを順番に実行>
Else If <条件=TRUE>
  __<テスト ステップを順番に実行>
Else
  __<テスト ステップを順番に実行>
End If
```

たとえば、以下の判断ロジックの条件では、変数 *propertyVar* とリテラル値「4」を比較しています。変数 *propertyVar* がリテラル値「4」より大きい場合は、ステップ 5 から 9 が実行されます。「4」より小さい場合は、ステップ 10 から 14 が実行されます。「4」と等しい場合は、ステップ 15 が実行されます。

```
If propertyVar より大きい 4
  __ステップ 5 からステップ 9 まで実行
Else If propertyVar より小さい 4
  __ステップ 10 からステップ 14 まで実行
Else
  __ステップ 15 を実行
End If
```

判断ロジックは、**テストロジックデザイナー** ウィザード、ロジックツールボックス、または If...Else If...Else テスト ステップを使用して作成できます。

ビジュアルテストへの判断ロジックの追加

このトピックでは、**テストロジックデザイナー** を使用して、ビジュアルテストに判断ロジックを追加する方法について説明します。判断ロジックテストステップをビジュアルテストに直接追加する方法については、「[判断テストステップを使用した判断ロジックの追加](#)」を参照してください。

1. ビジュアルテストを開きます。
2. テストロジックに含める 1 つのテストステップまたは一連のテストステップで右クリックします。
3. **挿入 > テストロジック > 判断** を選択します。**テストロジックデザイナー** ウィザードが開きます。
4. **次へ** をクリックします。
5. **ロジックの種類**の選択 ページで、判断ロジックの種類を選択します。
 - **コントロールのプロパティ** : テストアプリケーションのコントロールのプロパティに基づいて、一連のステップを実行する判断ロジックを作成します。例：

```
If propertyValue より大きい 2 ステップ 3 からステップ 4 まで実行 End If
```
 - **コントロールの存在確認** : 指定したコントロールが存在するかないかに基づいて、一連のステップを実行する判断ロジックを作成します。例：

```
"Button" が存在する 等しい True ステップ 3 からステップ 4 まで実行 End If
```
 - **このビジュアルテストで定義した変数** : ビジュアルテストに定義されているローカル変数の内容に基づいて一連のステップを実行する判断ロジックを作成します。例：

```
If stringVar = "red" ステップ 3 からステップ 4 まで実行 End If
```

- **高度なロジック** : ウィザードの条件の定義ステップをスキップし、**条件デザイナー** を使用して、詳細な判断ロジックを作成します。例 :

```
If stringVar = "red" および numberVar = 10 ステップ 3 からステップ 4 まで実行 End If
```

6. **次へ** をクリックします。

7. **コントロールのプロパティ** を選択した場合、**プロパティに基づく条件の定義** ページが開きます。

- a) **コントロール名** フィールドの右側にある **コントロールの識別** メニューを展開します。
- b) **コントロールの識別** メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション** : テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー** : テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ** : 非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
- c) **プロパティの選択** テーブルで、プロパティを選択します。
- d) **条件の選択** リストから、コントロールのプロパティに適用する条件ロジックを選択します。
- e) **期待値** フィールドには、プロパティに期待する値を指定します。
- f) **次へ** をクリックします。

8. **コントロールの存在確認** を選択した場合、**コントロールの存在条件の定義** ページが開きます。

- a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション** : テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー** : テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ** : 非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
- b) コントロールが存在することを確認する場合は、**コントロールが存在する** を選択します。
- c) コントロールが存在しないことを確認する場合は、**コントロールが存在しない** を選択します。
- d) **タイムアウト** フィールドには、コントロールが表示される、または表示されなくなるまで Silk Test Workbench が待機する時間を指定します。
デフォルトのタイムアウトは 0 で、コントロールが表示される、または表示されなくなるまで Silk Test Workbench は待機しないことを意味します。
- e) **次へ** をクリックします。

9. **このビジュアル テストで定義した変数** を選択した場合、**変数に基づく条件の定義** ページが開きます。

- a) **変数の選択** リストから事前に定義されたローカル変数を選択します。
このリストに変数を表示するには、ビジュアル テストにローカル変数を含める必要があります。ビジュアル テストに新しいローカル変数を追加する場合は、**新規作成** をクリックします。
評価結果 フィールドには、選択した変数のデータ型が表示されます。
- b) **条件の選択** から変数に適用する比較演算子を選択します。
- c) **期待値** フィールドには、変数に期待する値を指定します。
- d) **次へ** をクリックします。

10 **高度なロジック** を選択した場合、**次へ** をクリックします。

11 **判断の構築** ページが開きます。条件ロジック、および条件ロジックの結果が真となる場合に実行する、一連のステップを指定します。



ヒント: ポインタをリンクの上に置くと、完全なリンクの説明を含むツール ヒントが表示されます。

- a) **条件デザイナー** で条件ロジックを指定するには、**If** 文の右側にあるリンクをクリックします。
- b) 実行する一連のステップの最初のステップを指定するには、**ステップの開始** の右側にあるリンクをクリックします。
- c) 実行する一連のステップの最後のステップを指定するには、**ステップの終了** の右側にあるリンクをクリックします。
- d) 追加の条件ロジックを挿入するには、**Else If の追加** をクリックします。
用意されたリンクをクリックして **条件デザイナー** を開き、条件が True であるときに実行する一連のステップを指定します。
- e) 追加の条件ロジックを挿入するには、**Else の追加** をクリックします。
用意されたリンクを使用して、条件が False であるときに実行する一連のステップを指定します。
- f) 追加の条件ロジックおよび一連のステップを削除するには、**削除** をクリックします。
このボタンは、追加の条件ロジックを挿入したあとにのみ表示されます。

12次へ をクリックします。**要約** ページに、定義したテスト ロジックが表示されます。このページの最下部にあるチェック ボックスをオフにすると、**テスト ロジック デザイナ** ウィザードを今後使用するとき、**要約** ページが表示されなくなります。このチェック ボックスのオン/オフを切り替えると、**テスト ロジック デザイナの要約画面の表示** 全般オプションが更新されます。詳細については、「全般オプションの変更」を参照してください。

13完了 をクリックして、ビジュアル テストのテスト ステップにテスト ロジックを挿入します。テスト ロジックを変更するには、**戻る** をクリックして、必要な変更を加えます。テスト ロジックがビジュアル テストに追加されます。

判断テスト ステップを使用した判断ロジックの追加

If、Else If、および Else テスト ステップを挿入することですばやく判断ロジックを作成できます。これらのテスト ステップを使用すると、**テスト ロジック デザイナ** ウィザードとは別の方法で判断ロジックを作成できます。また、**テスト ロジック デザイナ** ウィザードを使用しなくても、既存の判断ロジックをすばやく変更できます。

Else If および Else テスト ステップでは、任意のレベル数にネストした複雑な判断ロジックを作成できます。

```
If <条件=TRUE>
  __<テスト ステップを順番に実行>
Else If <条件=TRUE>
  __<テスト ステップを順番に実行>
Else
  __<テスト ステップを順番に実行>
End If
```

たとえば、以下の判断ロジックの条件では、変数 *propertyVar* とリテラル値「4」を比較しています。変数 *propertyVar* がリテラル値「4」より大きい場合は、ステップ 5 から 9 が実行されます。「4」より小さい場合は、ステップ 10 から 14 が実行されます。「4」と等しい場合は、ステップ 15 が実行されます。

```
If propertyVar より大きい 4
  __ステップ 5 からステップ 9 まで実行
Else If propertyVar より小さい 4
  __ステップ 10 からステップ 14 まで実行
Else
  __ステップ 15 を実行
End If
```

ビジュアル テストに判断テスト ステップをすばやく挿入するには、以下を実行します。

1. ビジュアル テストを開き、判断ロジックに含めるテスト ステップを選択します。

一連のテスト ステップを選択するには、最初のステップを選択し、Shift キーを押しながら最後のテスト ステップを選択します。最初と最後のテスト ステップの間のテスト ステップはすべて自動的に選択されます。


2. ツールバーで **If タイプのロジック項目の挿入** をクリックします。

選択したテスト ステップが、If テスト ステップと End If テスト ステップで囲まれます。

3. 省略可能 : Else If または Else テスト ステップを If...End If ロジックに挿入するには、Else If または Else テスト ステップを挿入する場所の直前のテスト ステップを選択し、**Else If タイプのロジック項目の挿入** または **Else タイプのロジック項目の挿入** をクリックします。

判断ロジック テスト ステップの編集

テスト ロジック デザイナー ウィザードまたは **条件デザイナー** を使用してテスト ロジック ステップを作成したあと、**プロパティ** ペインから直接そのテスト ステップを編集できます。

 **注:** **条件** プロパティは、If および Else If テスト ステップにのみ含まれます。Else および End If テスト ステップには含まれません。

1. 編集するテスト ステップが含まれているビジュアル テストを開きます。

2. ビジュアル テストの **テスト ステップ** ウィンドウで、編集するテスト ステップを選択します。選択したテスト ステップのプロパティおよび関連するプロパティ値が、**プロパティ** ペインにプロパティ カテゴリごとに表示されます。

3. 使用可能な条件プロパティまたはその他のプロパティを編集します。

繰り返しロジック

繰り返しロジックは、指定した回数、またはユーザー定義の条件が true になるまで、一連のテスト ステップを繰り返します。繰り返しロジックで使用される条件は、2 つの値を比較し、その結果に基づいて値を返します。詳細については、「条件デザイナー」を参照してください。条件内の値には、テスト アプリケーション内のコントロールのプロパティの値、ビジュアル テストで使用される変数の値、リテラル値、またはアクティブ データ資産からの値のいずれかを使用できます。

指定した回数だけ一連のテスト ステップを繰り返す繰り返しロジックは、以下の要素で構成されます。

```
このループを <回数> 回繰り返します
__<テスト ステップを順番に実行>
繰り返しの終了
```

たとえば、以下の繰り返しロジックでは、テスト ステップのステップ 10 からステップ 15 までが繰り返されます。

```
このループを 10 回繰り返します
__ステップ 10 から
__ステップ 15 まで実行
繰り返しの終了
```

ユーザー定義の条件が True の間、または True になるまで一連のテスト ステップを繰り返す繰り返しロジックは、以下の要素で構成されます。

```
<条件=TRUE> の間繰り返します
__<テスト ステップを順番に実行>
繰り返しの終了
```

または、以下のようにループの最後に条件を指定できます。

```
繰り返します
__<テスト ステップを順番に実行>
<条件=TRUE> の間繰り返して終了
```

たとえば、以下の繰り返しロジックでは、変数 *propertyVar* の値がリテラル値「4」より小さい間は、一連のテスト ステップが繰り返されます。*propertyVar* が「4」より大きいか等しくなると、ループは実行されません。

```
propertyVar より小さい 4 の間繰り返します
__ステップ 10 から
__ステップ 14 まで実行
繰り返しの終了
```


または、ループの最後に条件を指定できます。以下の場合、`propertyVar` が「4」より大きいか等しいと、ループが 1 回実行されます。

```
繰り返します
__ステップ 10 から
__ステップ 14 まで実行
propertyVar より小さい 4 の間繰り返して終了
```

繰り返しロジックは、**テスト ロジック デザイナ** ウィザードまたはロジック ツールボックスを使用して作成できます。

ビジュアル テストへの繰り返しロジックの追加

このトピックでは、**テスト ロジック デザイナ** を使用して、ビジュアル テストに繰り返しロジックを追加する方法について説明します。

1. ビジュアル テストを開きます。
2. 繰り返したい 1 つのテスト ステップまたは一連のテスト ステップで右クリックします。
3. **挿入 > テスト ロジック > 繰り返し** を選択します。**テスト ロジック デザイナ** ウィザードが開きます。
4. **次へ** をクリックします。
5. **ロジックの種類を選択** ページで、繰り返しロジックの種類を選択します。

以下のオプションに基づいて、繰り返しロジックのタイプを選択します。

- **コントロールのプロパティ** : テスト アプリケーションのコントロールのプロパティに基づいて、一連のステップを繰り返す繰り返しロジックを作成します。例：
propertyValue 等しい 3 の間繰り返します __ステップ 3 から __ステップ 4 まで実行 繰り返しの終了
- **コントロールの存在確認** : 指定したコントロールが存在するかないかに基づいて、一連のステップを繰り返す繰り返しロジックを作成します。例：
"Button" が存在する 等しい True の間繰り返します __ステップ 3 から __ステップ 4 まで実行 繰り返しの終了
- **このビジュアル テストで定義した変数** : ビジュアル テストに定義されているローカル変数の内容に基づいて一連のステップを繰り返す繰り返しロジックを作成します。例：
numberVar より小さい 5 の間繰り返します __ステップ 3 から __ステップ 5 まで実行 繰り返しの終了
- **ステップ シーケンスを設定した回数繰り返す** : 一連のステップを設定した回数繰り返す繰り返しロジックを作成します。例：
このループを 10 回繰り返します __ステップ 3 から __ステップ 7 まで実行 繰り返しの終了
- **リストまたはコンボ ボックスの内容** : リスト ボックスまたはコンボ ボックス内の各項目に対して、一連のステップを繰り返す繰り返しロジックを作成します。例：
ListA の各項目ごとに繰り返します __ステップ 3 から __ステップ 5 まで実行 繰り返しの終了
- **アクティブ データ ファイルを使用してステップ シーケンスを繰り返す** : ビジュアル テストで使用するアクティブ データ ファイルのデータに基づいて、一連のステップを繰り返す繰り返しロジックを作成します。例：
アクティブ データ testData を使用して繰り返します __ステップ 3 から __ステップ 5 まで実行 繰り返しの終了
- **高度なロジック** : ウィザードの条件の定義 ステップをスキップし、**条件デザイナ** を使用して、詳細な繰り返しロジックを作成します。例：
propertyValue = red または stringVar = red の間繰り返します __ステップ 3 から __ステップ 5 まで実行 繰り返しの終了

6. **次へ** をクリックします。
7. **コントロールのプロパティ** を選択した場合、**プロパティに基づく条件の定義** ページが開きます。

- a) **コントロール名** フィールドの右側にある **コントロールの識別** メニューを展開します。
- b) **コントロールの識別** メニューを展開し、次のいずれかを選択します。
- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
- c) **プロパティの選択** テーブルで、プロパティを選択します。
- d) **条件の選択** リストから、コントロールのプロパティに適用する条件ロジックを選択します。
- e) **期待値** フィールドには、プロパティに期待する値を指定します。
- f) **次へ** をクリックします。
8. **コントロールの存在確認** を選択した場合、**コントロールの存在条件の定義** ページが開きます。
- a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。
- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
- b) コントロールが存在することを確認する場合は、**コントロールが存在する** を選択します。
- c) コントロールが存在しないことを確認する場合は、**コントロールが存在しない** を選択します。
- d) **タイムアウト** フィールドには、コントロールが表示される、または表示されなくなるまで Silk Test Workbench が待機する時間を指定します。
デフォルトのタイムアウトは 0 で、コントロールが表示される、または表示されなくなるまで Silk Test Workbench は待機しないことを意味します。
- e) **次へ** をクリックします。
9. このビジュアル テストで定義した変数 を選択した場合、**変数に基づく条件の定義** ページが開きます。
- a) **変数の選択** リストから事前に定義されたローカル変数を選択します。
このリストに変数を表示するには、ビジュアル テストにローカル変数を含める必要があります。ビジュアル テストに新しいローカル変数を追加する場合は、**新規作成** をクリックします。
評価結果 フィールドには、選択した変数のデータ型が表示されます。
- b) **条件の選択** から変数に適用する比較演算子を選択します。
- c) **期待値** フィールドには、変数に期待する値を指定します。
- d) **次へ** をクリックします。
10. **ステップ シーケンスを設定した回数繰り返す** を選択した場合、**繰り返し回数の定義** ページが開きます。
- a) 1 から 32767 の数値を入力して、一連のステップを繰り返す回数を指定します。
- b) **次へ** をクリックします。
11. **リストまたはコンボ ボックスの内容** を選択した場合、**コントロールに基づく繰り返しの定義** ページが開きます。
- a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。
- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。

- **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。

いずれのタイプのコントロールでも、リスト ボックスまたはコンボ ボックスの各項目の値を使用する繰り返しロジックを作成できます。たとえば、リスト ボックスまたはコンボ ボックスの値に対して反復処理を行い、各項目の値をテスト アプリケーションのテキスト ボックスに挿入する繰り返しループを作成できます。このようにして、リスト ボックスまたはコンボ ボックスの内容全体をすばやくテストできます。

b) **次へ** をクリックします。

12 アクティブ データ ファイルを使用してステップ シーケンスを繰り返す を選択した場合、使用するアクティブ データ資産の定義 ページが開きます。

使用するアクティブ データ資産の定義 ページを使用して、繰り返しロジックで使用するアクティブ データ資産とアクティブ データ資産内のデータの取得方法を定義します。

a) リストから **アクティブ データ資産** を選択します。

アクティブ データ資産をこのリストに表示するためには、まずビジュアル テストにアクティブ データ資産を含める必要があります。

b) 使用するデータ ファイルのシートを、**シート名** リストから選択します。

デフォルトでは、アクティブ データ資産で指定されているシートが使用されます。

c) 使用するアクティブ データ ファイルの最初のデータ行の番号を、**開始行** フィールドに入力します。

d) アクティブ データ ファイルの最後のデータ行から開始するには、**データを含む最後の行から開始** チェック ボックスをオンにします。

ファイルの最後の行が Silk Test Workbench によって自動的に検出され、アクティブ データ ファイルの行に対し逆方向で反復処理が行われます。たとえば、アクティブ データ ファイルに 10 行が含まれているときにこのオプションを選択した場合は、行 10、行 9、行 8 という順番で反復処理が Silk Test Workbench によって行われます。

e) 使用するアクティブ データ ファイルの最後のデータ行の番号を、**終了行** フィールドに入力します。

f) アクティブ データ ファイルの最後のデータ行で終了するには、**データを含む最後の行で終了** チェック ボックスをオンにします。

ファイルの最後の行が Silk Test Workbench によって自動的に検出されます。



ヒント: **データを含む最後の行から開始** チェック ボックスまたは **データを含む最後の行で終了** がオンになっている場合は、繰り返し ステップの Start Row プロパティまたは End Row プロパティの値が -1 に設定されます。

g) 行が処理される順番を選択します。

- **順序通りにすべての行を取得**：アクティブ データ ファイルで、選択した行の最初に指定した行から最後に指定した行まで順番に反復処理を行うには、このオプションを選択します。
- **ランダムな順序ですべての行を取得**：アクティブ データ ファイルのすべての行でランダムに反復処理を行うには、このオプションを選択します。
- **ランダムに X 行を取得**：指定した行数に対してランダムに反復処理を行うには、このオプションを選択してアクティブ データ ファイルの行数を入力します。

h) **次へ** をクリックします。

13 高度なロジック を選択した場合、**次へ** をクリックします。

14 繰り返しの構築 ページが開きます。繰り返す一連のステップを指定します。



ヒント: ポインタをリンクの上に置くと、完全なリンクの説明を含むツール ヒントが表示されます。

a) 繰り返し期間では、ループの開始または終了時に確認される繰り返し条件を選択します。

b) 繰り返し期間に対して、**条件デザイナー** で条件ロジックを指定するには、**繰り返し期間** 文の右側にあるリンクをクリックします。

c) 繰り返す一連のステップの最初のステップを指定するには、**ステップの開始** の右側にあるリンクをクリックします。

d) 繰り返す一連のステップの最後のステップを指定するには、**ステップの終了**の右側にあるリンクをクリックします。

15次へ をクリックします。要約 ページに、定義したテスト ロジックが表示されます。このページの最下部にあるチェック ボックスをオフにすると、**テスト ロジック デザイナ** ウィザードを今後使用するとき、**要約** ページが表示されなくなります。このチェック ボックスのオン/オフを切り替えると、**テスト ロジック デザイナの要約画面の表示** 全般オプションが更新されます。詳細については、「全般オプションの変更」を参照してください。

16完了 をクリックして、ビジュアル テストのテスト ステップにテスト ロジックを挿入します。テスト ロジックを変更するには、**戻る** をクリックして、必要な変更を加えます。テスト ロジックがビジュアル テストに追加されます。

リスト ボックスまたはコンボ ボックスの内容に基づいた繰り返しロジックの作成

繰り返しロジックで使用するリスト ボックス コントロールまたはコンボ ボックス コントロールを特定します。いずれのタイプのコントロールでも、リスト ボックスまたはコンボ ボックスの各項目の値を使用する繰り返しロジックを作成できます。たとえば、リスト ボックスまたはコンボ ボックスの値に対して反復処理を行い、各項目の値をテスト アプリケーションのテキスト ボックスに挿入する繰り返しループを作成できます。このようにして、リスト ボックスまたはコンボ ボックスの内容全体をすばやくテストできます。

1. ビジュアル テストを記録します。
2. リスト ボックスまたはコンボ ボックス コントロールに基づいて繰り返しロジックを作成するには、**テスト ロジック** ウィザードを使用します。
3. リテラル値を予約変数 `st_RepeatListValue` に置き換えて、リスト ボックスまたはコンボ ボックスの内容を使用する繰り返しテスト ステップを変更します。
予約変数 `st_RepeatListValue` は、繰り返しループを反復するたびに、リスト ボックスまたはコンボ ボックスの各項目の値を格納するために使用します。`st_RepeatListValue` は、項目のインデックスとテキストの両方が含まれている複合値を返します。
4. テスト ステップを変更するには、繰り返すテスト ステップを選択します。
5. **プロパティ** ウィンドウで、変更するテスト ステップの自動化プロパティを選択します。
6. **選択** をクリックして、**変数** を選択します。 **変数の選択** ダイアログ ボックスが開きます。
7. **変数の種類** リストから **予約** を選択し、**st_RepeatListValue** を選択します。
8. **OK** をクリックします。Silk Test Workbench によって、テスト ステップのテキストおよびプロパティ 値が、選択した変数名に更新されます。
9. ビジュアル テストを実行します。

繰り返しロジック テスト ステップの編集

テスト ロジック デザイナ ウィザードまたは **条件デザイナ** を使用してテスト ロジック ステップを作成したあと、**プロパティ** ペインから直接そのテスト ステップを編集できます。

1. 編集するテスト ステップが含まれているビジュアル テストを開きます。
2. ビジュアル テストの **テスト ステップ** ウィンドウで、編集する繰り返しロジック テスト ステップを選択します。選択したテスト ステップのプロパティおよび関連するプロパティ値が、**プロパティ** ペインにプロパティ カテゴリごとに表示されます。
3. 使用可能な条件プロパティまたはその他のプロパティを編集します。
繰り返し テスト ステップには、条件プロパティとその他のプロパティの両方が含まれています。繰り返しの終了には、その他のプロパティのみが含まれています。

検証ロジック

検証ロジックは、ユーザー定義の条件を評価して、合格/失敗メッセージをビジュアル テストまたはスクリーンショットの結果に送信します。必要に応じて、失敗のテスト ステップにフラグを付けて、ビジュアル テストの結果に表示することもできます。検証ロジックで使用される条件は、2つの値を比較し、その結果に基

づいて値を返します。詳細については、「条件デザイナ」を参照してください。条件内の値には、テストアプリケーション内のコントロールのプロパティの値、ビジュアルテストで使用される変数の値、または一連のテストステップの実行にかかる時間のいずれかを使用できます。

検証ロジックは、以下の要素で構成されます。

```
If <条件=TRUE>
  __<合格メッセージを結果に送信>
Else
  __<失敗メッセージとフラグを結果に送信>
End If
```

たとえば、以下の判断ロジックの条件では、変数 *propertyVar* とリテラル値「4」を比較しています。変数 *propertyVar* がリテラル値「4」より大きい場合は、合格メッセージが結果に送信されます。値が「4」より小さい場合は、失敗メッセージとフラグが送信されます。

```
If propertyVar より大きい 4
  __<合格メッセージを送信>
Else
  __<失敗メッセージとフラグを送信>
End If
```

検証ロジックは、**テストロジックデザイナ** ウィザードまたはロジックツールボックスを使用して作成できます。ビジュアルテストの記録中に検証ロジックを作成することもできます。

ビジュアルテストへの検証ロジックの追加

このトピックでは、**テストロジックデザイナ** を使用して、ビジュアルテストに検証ロジックを追加する方法について説明します。どの種類の検証ロジックも、ユーザー定義の合格/失敗メッセージを送信し、フラグをビジュアルテストの結果に設定します（任意）。

1. ビジュアルテストを開きます。
2. 検証したいオブジェクトを含むテストステップで右クリックします。
3. **挿入 > テストロジック > 検証** を選択します。**テストロジックデザイナ** ウィザードが開きます。
4. **次へ** をクリックします。
5. **ロジックの種類**の選択 ページで、検証ロジックの種類を選択します。

以下のオプションに基づいて、検証ロジックのタイプを選択します。

- **コントロールのプロパティ**：コントロールのプロパティの値に基づいて検証ロジックを作成します。例：
「"Button"."Text" 等しい "Cancel"」を検証します
- **コントロールの存在確認**：指定したコントロールが存在するかどうかを検査する検証ロジックを作成します。例：
「"Button" が存在する 等しい True」を検証します
- **このビジュアルテストで定義した変数**：ビジュアルテストに定義されているローカル変数の内容に基づいて検証ロジックを作成します。例：
「st_LastValidation 等しい False」を検証します
- **Rumba 画面のコンテンツ**：Rumba 画面のコンテンツを比較するための検証ロジックを作成します。
- **検証資産**：UI のコンテンツを資産と比較するための検証ロジックを作成します。現時点では、Silk Test Workbench は、検証資産としてイメージ検証のみをサポートしています。詳細については、「資産の種類」を参照してください。例：
'Untitled_1' という名前のイメージ検証を実行します
- **ステップシーケンスの実行にかかる時間**：ステップシーケンスの実行にかかる時間に基づいて検証ロジックを作成します。例：
「タイマー番号 '1' が 0 から 2 秒の間にある」を検証します __ステップ 3 から __ステップ 4 を実行します

6. **次へ** をクリックします。

7. **コントロールのプロパティ** を選択した場合、**プロパティに基づく条件の定義** ページが開きます。

a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。

- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
- **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
- **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。

b) さらに、**コントロール名** フィールドにコントロールの完全ロケータを入力して、**更新** をクリックすることもできます。テスト対象アプリケーションでコントロールが見つかったら、**プロパティの選択** リストがコントロールのプロパティで更新されます。

c) **プロパティの選択** テーブルで、プロパティを選択します。

d) **条件の選択** リストから、コントロールのプロパティに適用する条件ロジックを選択します。

e) **期待値** フィールドには、プロパティに期待する値を指定します。

f) **次へ** をクリックします。

8. **コントロールの存在確認** を選択した場合、**コントロールの存在条件の定義** ページが開きます。

a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。

- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
- **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
- **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。

b) コントロールが存在することを確認する場合は、**コントロールが存在する** を選択します。

c) コントロールが存在しないことを確認する場合は、**コントロールが存在しない** を選択します。

d) **タイムアウト** フィールドには、コントロールが表示される、または表示されなくなるまで Silk Test Workbench が待機する時間を指定します。

デフォルトのタイムアウトは0で、コントロールが表示される、または表示されなくなるまで Silk Test Workbench は待機しないことを意味します。

e) **次へ** をクリックします。

9. このビジュアル テストで定義した変数 を選択した場合、**変数に基づく条件の定義** ページが開きます。

a) **変数の選択** リストから事前に定義されたローカル変数を選択します。

このリストに変数を表示するには、ビジュアル テストにローカル変数を含める必要があります。ビジュアル テストに新しいローカル変数を追加する場合は、**新規作成** をクリックします。

評価結果 フィールドには、選択した変数のデータ型が表示されます。

b) **条件の選択** から変数に適用する比較演算子を選択します。

c) **期待値** フィールドには、変数に期待する値を指定します。

d) **次へ** をクリックします。

10 Rumba 画面のコンテンツ を選択した場合、Rumba 画面上のコントロールを識別します。

a) **コントロールの識別** メニューを展開し、次のいずれかを選択します。

- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
- **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。

- **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。

b) **次へ** をクリックします。

11 検証資産 を選択した場合、**検証資産の構築** ページが開きます。

- a) **新しい検証の作成** をクリックすると、**イメージ検証** UI が開きます。この UI で、新しい検証資産を作成できます。
- b) **既存の検証の挿入** をクリックすると **検証の参照** ダイアログ ボックスが開きます。このボックスを使用して、テストで実行する検証資産を選択できます。
- c) 省略可能：**検証の失敗時にスクリーンショットをキャプチャする** チェック ボックスをオンにすると、検証の失敗時にスクリーンショットが結果ファイルに追加されます。
- d) **次へ** をクリックします。

12 ステップ シーケンスの実行にかかる時間 を選択した場合、**タイミングに基づく条件の定義** ページが開きます。一連のテスト ステップと、それを再生する時間の範囲を指定できます。

- a) 検証する一連のステップの最初のステップを指定するには、**ステップの開始** の右側にあるリンクをクリックします。
- b) 検証する一連のステップの最後のステップを指定するには、**ステップの終了** の右側にあるリンクをクリックします。
- c) **許容時間範囲** の 2 つのフィールドの値を定義して、一連のステップの再生にかかる時間の範囲を設定します。

たとえば、時間の範囲を 10 から 30 秒に設定した場合、指定した一連のステップの実際の再生時間が 10 秒未満または 30 秒を超える場合は、ビジュアル テストの結果に失敗のメッセージが送信されます。一方、実際の再生時間が 10 から 30 秒の場合は、ビジュアル テストの結果に合格のメッセージが送信されます。

d) **タイマー番号の使用** リストからタイマーを選択します。

複数の条件の時間を計るときに、各タイマーに一意の番号を割り当ててタイマーを区別できます。タイマーを使用して検証ロジックを作成すると、**テスト ステップ** ウィンドウに 2 つのテスト ステップが表示されます。最初のテスト ステップには **タイマー番号 X の開始** というラベルが付けられ (X は特定のタイマー番号)、検証する一連のステップの最初のテスト ステップの直前に表示されます。2 番めのテスト ステップには **タイマー番号 X の停止** というラベルが付けられ、検証する一連のステップの最後のステップの直後に表示されます。



ヒント：検証するステップの順序を変更する場合は、**テスト ステップ** ウィンドウで、いずれかのタイマーのテスト ステップを切り取り、目的の位置に貼り付けます。

e) **次へ** をクリックします。

13 検証の構築 ページが開きます。ビジュアル テストの結果に表示する合格/失敗のメッセージを作成します。また、検証が失敗した場合に、結果および現在のユーザーの **開始画面** にフラグを送信することもできます。

- a) 省略可能：**If** 文の右側にある条件リンクをクリックして、**条件デザイナー** を開き、条件を編集します。
- b) **結果の詳細に以下の '合格テキスト' を表示する** フィールドに、再生時に検証が合格したときに結果に表示するテキストを入力します。
このテキストは、**詳細** タブの **テスト ステップ** ペインで、検証ステップの **結果の詳細** 列に表示されます。
- c) **そうでなければ、結果の詳細に以下の '失敗テキスト' を表示する** フィールドに、再生時に検証が失敗したときに結果に表示するテキストを入力します。
このテキストは、**失敗** タブ、**フラグ** タブ、および **詳細** タブの **テスト ステップ** ペインで、検証ステップの **結果の詳細** 列に表示されます。
- d) 省略可能：**検証に失敗した場合に、結果に 'フラグ' を作成する** チェック ボックスをオンにすると、検証が失敗したときにフラグが作成されます。

このフラグは、**失敗** タブ、**フラグ** タブ、および **詳細** タブの **テスト ステップ** ペインで、検証ステップの **結果の詳細** 列に表示されます。デフォルトでは、このフラグは現在のユーザーに自動的に割り

当てられ、現在のユーザーの **開始画面** の **フラグ** ペインに表示されます。このフラグを別の人に割り当てる場合は、再生後に結果から再割り当てする必要があります。

- e) 省略可能：**以下のフラグ説明を表示する** フィールドに検証が失敗したときのフラグの説明を入力します。
- f) 省略可能：**検証の失敗時にスクリーンショットをキャプチャする** チェックボックスをオンにすると、検証の失敗時にスクリーンショットが結果ファイルに追加されます。

14次へ をクリックします。**要約** ページに、定義したテスト ロジックが表示されます。このページの最下部にあるチェックボックスをオフにすると、**テスト ロジック デザイナ** ウィザードを今後使用するとき、**要約** ページが表示されなくなります。このチェックボックスのオン/オフを切り替えると、**テスト ロジック デザイナの要約画面の表示** 全般オプションが更新されます。詳細については、「**全般オプションの変更**」を参照してください。

15完了 をクリックして、ビジュアルテストのテストステップにテスト ロジックを挿入します。テスト ロジックを変更するには、**戻る** をクリックして、必要な変更を加えます。テスト ロジックがビジュアルテストに追加されます。

検証ロジック テスト ステップの編集

1. 編集するテストステップが含まれているビジュアルテストを開きます。
2. ビジュアルテストの **テストステップ** ウィンドウで、編集する検証ロジック テストステップを選択します。選択したテストステップのプロパティおよび関連するプロパティ値が、**プロパティ** ペインにプロパティ カテゴリごとに表示されます。
3. 使用可能な条件プロパティ、検証結果の説明プロパティ、フラグの設定プロパティ、またはその他のプロパティを編集します。

エラー処理ロジック

エラー処理ロジックを設定すると、再生エラーをデバッグする必要がなくなります。エラー処理を設定するには、ビジュアルテストでエラーの検出ステップとその処理方法を設定します。エラー処理を設定することによって、ビジュアルテストを最後まで再生でき、再生中に発生したエラーの情報を得ることができます。

エラー処理ロジックは、以下の要素で構成されます。

```
If <エラー=TRUE>  
  __<実行する操作>
```

たとえば、エラー処理で次のステップから再開するように設定していないかぎり、エラーが発生すると、失敗メッセージとフラグが送信されます。

エラー時に '再開して次に進む'

エラー処理ロジックを作成するには、**テスト ロジック デザイナ** ウィザードまたはロジック ツールボックスを使用するか、**挿入 > エラー処理** を選択します。

ビジュアルテストへのエラー処理ロジックの追加

このトピックでは、**テスト ロジック デザイナ** を使用して、ビジュアルテストにエラー処理ロジックを追加する方法について説明します。

1. ビジュアルテストを開きます。
2. エラー処理ロジックを追加する場所のテストステップを右クリックします。
3. **挿入 > テスト ロジック > エラー処理** を選択します。**テスト ロジック デザイナ** が開きます。
4. **次へ** をクリックします。**ロジックの種類を選択** ページが開きます。
5. **次へ** をクリックします。**エラー処理の構築** ページが開きます。
6. エラー処理ロジックを指定して、エラーが発生した場合に使用するには、リストから次の項目のいずれかを選択します。

デフォルトの再生エラー ダイアログボックスを表示します **再生エラー** ダイアログボックスを表示し、そこでそのエラーに対して実行する操作を決定します。**再生エラー** ダイアログボックスを使用してデバッグモードに入り、ビジュアルテストでエラーを診断します。

現在のビジュアルテストの再生を終了します エラーが発生したときに、ビジュアルテストの再生を終了します。ビジュアルテストのその他のステップは再生されません。再生が終了すると、**再生完了** ダイアログボックスが表示され、そこで再生の完了時に実行する操作を決定します。

エラーが発生したステップの再生を再試行します **資産** プロパティに指定されているビジュアルテストを再生してから、エラーが発生したステップの実行を試行します。**資産** プロパティにビジュアルテストが指定されていない場合、単にエラーが発生したステップの再実行を試行します。**再試行の回数** プロパティには、再生を試行する回数を指定できます。デフォルト値は、5です。指定した再試行の回数に到達した時点でステップ実行に失敗すると、再生エラーが生成されます。

次のステップで再生を再開します **資産** プロパティに指定されているビジュアルテストを再生してから、エラーが発生したステップの直後のステップの実行を試みます。エラーの原因となったステップは、再実行されません。

ラベルへ移動します **資産** プロパティに指定されているビジュアルテストを再生し、**ラベル** プロパティに指定されている事前定義のラベルステップに移動して、再生を続行します。**移動** を使用すると、ビジュアルテストで、再生中に発生したエラーと同様のエラーが発生する可能性のあるステップをスキップして、テストの別のセクションに移動できます。

他のビジュアルテストを最初に再生します 指定した別のビジュアルテストが再生されたあと、次に実行される操作をユーザーが指定します。別のテスト実行のあとに実行される操作を決定するには、テストの終了、エラーが発生したステップを再試行、ラベルへ移動、またはエラーが発生した元のテストの再開を選択します。

7. **次へ** をクリックします。**要約** ページに、定義したテストロジックが表示されます。このページの最下部にあるチェックボックスをオフにすると、**テストロジックデザイナー** ウィザードを今後使用するとき、**要約** ページが表示されなくなります。このチェックボックスのオン/オフを切り替えると、**テストロジックデザイナーの要約画面の表示** 全般オプションが更新されます。詳細については、「全般オプションの変更」を参照してください。

8. **完了** をクリックして、ビジュアルテストのテストステップにテストロジックを挿入します。テストロジックを変更するには、**戻る** をクリックして、必要な変更を加えます。テストロジックがビジュアルテストに追加されます。

エラー処理ロジックテストステップの編集

1. 編集するテストステップが含まれているビジュアルテストを開きます。
2. ビジュアルテストの **テストステップ** ペインで、編集するエラー処理ロジックテストステップを選択します。選択したテストステップのプロパティおよび関連するプロパティ値が、**プロパティ** ペインにプロパティカテゴリごとに表示されます。
3. 必要に応じて、プロパティを編集します。

テストロジックデザイナー

テストロジックデザイナー ウィザードウィザードでは、テストロジックを作成してビジュアルテストに挿入できます。このウィザードにアクセスするには、ビジュアルテストを開いているときに、**挿入 > テストロジック** メニューを選択します。また、ロジックツールボックスからもアクセスできます。その場合、さまざまなタイプのテストロジックについて把握し、目的のテストロジックを作成するために必要なウィザードの該当ページにすばやくアクセスできます。

以下のタイプのテストロジックを挿入するには、**テストロジックデザイナー** ウィザードを使用します。

- 判断ロジック

- 繰り返しロジック
- 検証ロジック
- エラー処理ロジック

テスト ロジック デザイナ ウィザードの最初のページは **ようこそ** ページで、ここにはウィザードのロジックステップのリスト、および設計する特定のタイプのテスト ロジックが表示されます。

このウィザードおよびそれ以外のすべての Silk Test Workbench ウィザードを今後使用するとき、**ようこそ** ページをスキップするには、このページの最下部にあるチェック ボックスをオンにします。このチェック ボックスをオンにすると、**全般** オプションの **ウィザードの [ようこそ] 画面の表示** オプションが更新されます。

ウィザードのサイズは、任意のページの右下隅をドラッグして変更できます。または、**ロジック ステップ** ペインのタイトルの横の矢印ボタンをクリックするか、**全般** オプションの **テスト ロジック デザイナのステップの表示** ペイン オプションを設定することでも、ウィザードの **ロジック ステップ** ペインを表示/非表示にできます。

テスト ロジック デザイナ ウィザードの最後のページは、**要約** ページです。このページの最下部にあるチェック ボックスをオンにすると、**テスト ロジック デザイナ** ウィザードを今後使用するとき、**要約** ページが表示されなくなります。このチェック ボックスのオン/オフを切り替えると、**テスト ロジック デザイナの要約画面の表示** 全般オプションが更新されます。詳細については、「全般オプションの変更」を参照してください。

ステップの選択

ステップの選択 ダイアログ ボックスを使用して、ビジュアル テストの単一のステップまたは一連のステップを選択できます。

1. ステップを選択します。
2. 省略可能：一連のステップを選択するには、Shift キーを押しながら最後のステップを選択します。
3. **OK** をクリックします。

ロジック ツールボックス

ロジック ツールボックスは、テスト ロジックをビジュアル テストにすばやく挿入できるナビゲーション支援機能です。ロジック ツールボックスでは、さまざまなタイプのテスト ロジックについて把握し、目的のテスト ロジックを選択してビジュアル テストに挿入できます。ロジック ツールボックスでテスト ロジックのタイプを選択すると、Silk Test Workbench によって **テスト ロジック デザイナ** ウィザードの該当ページが開き、テスト ロジックを作成して挿入できます。**このビジュアル テストの再生中に出現/消滅するオブジェクトを待機** を使用する場合は、ウィザードを使用せずに、**プロパティ** ペインでオブジェクトプロパティを指定します。

ロジック ツールボックスでは、一般的なタイプのテスト ロジックが以下のカテゴリに整理されます。

- **画面内容ロジック**: テスト アプリケーションの画面コントロールに基づいて一連のステップを実行する判断 (If) ロジックまたは繰り返しロジック。
- **ユーザー定義のフォローアップ ロジック**: ユーザー定義の合格/失敗メッセージとオプションのフラグをビジュアル テストの結果に送信する検証ロジック。また、指定したユーザーの開始画面にフラグを送信して、特定のテスト ステップまたはビジュアル テストの結果を通知することもできます。
- **エラー処理と同期**: ステップを実行する前に出現/消滅するオブジェクトを待機する同期ロジック。再生中にエラーが発生した場合、ユーザー定義の操作を実行するエラー処理ロジック。

ロジック ツールボックスの各カテゴリでは、各ロジック タイプの説明を参照し、目的のロジック タイプをダブルクリックして、ビジュアル テストにすばやく挿入できます。詳細については、「ロジック ツールボックスを使用したテスト ロジックの挿入」を参照してください。

ロジック ツールボックスを使用したテスト ロジックの挿入

ロジック ツールボックスを使用して、ビジュアル テストにテスト ロジックを挿入できます。

1. ビジュアル テストを開きます。

デフォルトでは、**ロジック ツールボックス** タブは **テスト ステップ** ペインの右側に表示されます。デフォルトでこのタブが表示されない場合は、**表示 > ロジック ツールボックス** をクリックします。

2. **ロジック ツールボックス** タブをクリックして、ロジック ツールボックスを開きます。



ヒント: ロジック ツールボックスのサイズを変更するには、ポインタを端に合わせて、目的の幅になるまでツールボックスをドラッグします。

3. ロジック カテゴリをクリックして、ビジュアル テストに挿入できる各カテゴリのさまざまなロジック タイプを表示します。

ロジック タイプの説明を表示するには、目的のロジック タイプをクリックします。説明は、ロジック ツールボックスの下部に表示されます。

4. ロジック タイプをダブルクリックすると、**テスト ロジック デザイナ** ウィザードの該当ページが自動的に開きます。

または、ロジック タイプを **テスト ステップ** ペインの適切な位置にドラッグ&ドロップすることもできます。

5. **テスト ロジック デザイナ** ウィザードを使用して、ビジュアル テストにテスト ロジックを作成および挿入します。

記録中のテスト ロジックの挿入

ビジュアル テストの記録中にテスト ロジックを挿入できます。

1. 記録を開始します。

2. テスト ロジックを追加するポイントに到達したとき、以下の手順のいずれかを実行します。

- **Alt+F9** を押します。

このショートカット キーの組み合わせを使用すると、記録が一時的に停止され、現在強調表示されているオブジェクトが検証のために選択されます。**テスト ロジック デザイナ** ウィザードが開き、選択されたオブジェクト セットが検証対象のコントロールとなります。

- **記録中** ダイアログ ボックスで、**検証の追加** をクリックするか、または **Ctrl+Alt** を押します。

このオプションを実行すると、記録が一時的に停止され、**テスト ロジック デザイナ** ウィザードが表示されます。ウィザードが開いたあと、検証するオブジェクトを選択します。

3. ウィザードの指示に従って処理を行い、**完了** をクリックしてウィザードを閉じ、記録を続行します。記録が完了すると、新しいテスト ロジックが **ビジュアル ナビゲータ** の **テスト ステップ** ペインに表示されます。

既存のビジュアル テストへのテスト ロジックの挿入

作成したビジュアル テストにテスト ロジックを挿入することができます。

1. ビジュアル テストを開きます。

2. テスト ロジックに含める 1 つのテスト ステップまたは一連のテスト ステップを選択します。

3. **挿入 > テスト ロジック** を選択し、適切なロジックの種類を選択します。

Ctrl+Alt を押して、ビジュアル テストに検証ロジックを追加することもできます。

テスト ロジック デザイナ ウィザードが開きます。

4. ウィザードに従って処理を行い、**完了** をクリックします。

テスト ステップの説明が長すぎて **テスト ステップ** ペインに表示できない場合は、ポインタをテスト ステップに合わせて、ツール ヒントとして説明全体が表示されます。

テスト ロジック テスト ステップがビジュアル テストに表示されます。

条件デザイナー

条件デザイナー ダイアログ ボックスでは、判断、繰り返し、および検証テスト ロジックの結果を導く条件をすばやく定義できます。**条件デザイナー** には、**テスト ロジック デザイナ** ウィザードまたはロジック

ツールボックスからアクセスします。特定のタイプのテスト ロジックを作成したあと、**プロパティ** ペインから **条件デザイナー** にアクセスして、条件を編集することもできます。

条件デザイナー を使用して定義される条件は、以下の要素で構成されます。

```
<値 1> <比較演算子> <値 2> = <戻り値>
```

例：

```
5 より大きい 4 = TRUE
```

条件の各要素は、以下のとおりです。

- 値 1 と値 2：条件の値は、互いに比較される入力パラメータとして使用されます。条件の値には、コントロールのプロパティ、変数、アクティブ データ資産、またはリテラル値を指定できます。
- 比較演算子：比較演算子は、条件の比較ロジックと、条件内の 2 つの値の相互関係を定義します。比較演算子を以下に示します。
 - 等しい (==)
 - 等しくない (!=)
 - より大きい (>)
 - より小さい (<)
 - 以上 (>=)
 - 以下 (<=)
 - 含む：値 1 のテキストに 値 2 のテキストが含まれる場合は True。それ以外の場合は False。
 - 含まない：値 1 のテキストに 値 2 のテキストが含まれない場合は True。それ以外の場合は False。
- 戻り値：比較演算子を使用して 2 つの値を比較したとき、評価された条件の戻り値は TRUE または FALSE のいずれかとなります。

各テスト ロジック タイプで、条件の戻り値を使用して実行される操作が決定されます。判断ロジックと繰り返しロジックの場合は、一連のテスト ステップが実行されます。検証ロジックの場合は、ビジュアル テストの結果に合格または失敗のメッセージが送信されるか、オプションで、指定されたテスト ステップにフラグが立てられます。

判断テスト ロジックの例

```
If counterVar 等しい 12  
ステップ 2 からステップ 4 まで実行  
End If
```

繰り返しテスト ロジックの例

```
counterVar 等しい 12 の間繰り返します  
ステップ 2 から  
ステップ 4 まで実行  
繰り返しの終了
```

検証テスト ロジックの例

```
If counterVar 等しい 12  
合格メッセージを送信  
else  
失敗メッセージを送信  
End If
```

また、論理演算子 AND または OR を使用して複合条件を作成できます。たとえば、以下のような複合の判断ロジック条件を作成できます。

```
If counterVar 等しい 12  
AND  
textVar 等しい "Admin"
```

ステップ 2 からステップ 4 まで実行
End If

条件デザイナーを使用した条件の定義

条件デザイナー ダイアログ ボックスを使用して、詳細なテスト ロジックの条件を定義または編集できます。

1. 条件デザイナー には **テスト ロジック デザイナー** ウィザードからアクセスします。


または、If テスト ステップを挿入し、**プロパティ** ウィンドウで **条件** プロパティを選択して、**条件デザイナー** をクリックします。


条件デザイナー が開きます。

2. 条件の定義

セクションの **値 1** テキスト ボックスで、最初の条件の値を指定します。この値には、コントロールのプロパティ、任意のタイプの変数、アクティブ データ資産、リテラル値、またはコントロールの存在確認を指定できます。

テキスト ボックスに直接値を入力するか、... をクリックして **条件の選択** ダイアログ ボックスにアクセスします。

 **ヒント:** 値を選択すると、その値の型が Silk Test Workbench によって **値 1** テキスト ボックスに表示されます。

 **注:** データ型は、Double から Long、テキスト、ブール値など、他の型に変換できます。ただし、このような変換によってデータがどのような影響を受けるかを確認することが重要です。データ型の変換の詳細については、「式デザイナーの関数の使用」を参照してください。

3. 演算子

リストから比較演算子を選択します。

比較演算子では、条件の比較ロジックと条件内の 2 つの値の相互関係を定義します。以下の演算子を利用できます。

- 等しい (==)
- 等しくない (!=)
- より大きい (>)
- より小さい (<)
- 以上 (>=)
- 以下 (<=)
- 含む : **値 1** のテキストに **値 2** のテキストが含まれる場合は True。それ以外の場合は False。
- 含まない : **値 1** のテキストに **値 2** のテキストが含まれない場合は True。それ以外の場合は False。

4. ステップ 2 で最初の値を指定した手順と同じ手順に従って、**値 2** テキスト ボックスで 2 番目の条件値を指定します。


5. 評価結果

リストから、条件の評価に対するデータ型を選択します。

使用可能なデータ型は、以下のとおりです。

テキスト

変数の値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

 **注:** テキスト データ型は大文字/小文字を区別します。つまり、比較時にはテキスト文字列の個々の文字の大文字/小文字が一致していなければなりません。大文字/小文字が一致していない場合は、比較は失敗します。

テキスト (大文字と小文字を区別しない)

評価中の文字列比較において、文字列の個々の文字の大文字小文字が一致しないことを除き、テキストは同じです。

数値 (Long)

変数の値は、-2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。

数値 (Double)	変数の値は、倍精度浮動小数点数で、-1.7E308 から +1.7E-307 の 64 ビットの数値として格納されます。
ブール値 (True/False)	変数の値は True または False です。
数値 (Long Long)	64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。
列挙	<p>この変数は一連の値をグループ化し、1 から n まで順番に番号を付けます。限られた個数の一意の値のみを変数に保持する場合は、列挙型を宣言します。この変数型は、列挙データ型を必要とするプロパティまたは変数に使用します。</p> <p>列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。</p>

- 定義した条件を **条件** テーブルに追加するには、**追加** をクリックします。
単一の条件ステートメントを作成する場合、**ロジック** リスト (AND、OR) は無視してください。**ロジック** ボックスは、複合条件を作成する場合にのみ使用します。
- 省略可能：**上へ移動** または **下へ移動** をクリックすると、**条件** テーブルで選択した条件を 1 行上に、または 1 行下に移動できます。
論理演算子 (AND、OR) を含む条件を **条件** テーブルの先頭行に移動した場合、論理演算子は条件と一緒に移動されません。条件ロジックの構造を維持するために、演算子は 2 行めに残ります。
- オプション：複合条件を作成する場合は、前のステップに従って追加条件を作成し、**ロジック** リストから結合ロジックのタイプを選択します。**追加** をクリックして、作成済みの条件のあとに条件を追加します。
- 省略可能：選択されている条件を **条件** テーブルから削除するには、**削除** を使用します。
- 省略可能：**プレビュー** をクリックすると、定義済みの条件が読みやすいテキスト形式で表示されます。

条件デザイナを使用した条件の編集

- ビジュアル テストのテスト ステップの **プロパティ** ペインで **条件** プロパティを選択し、**条件デザイナ** をクリックします。**条件デザイナ** が開きます。
- 既存の条件の値を編集するには、編集する条件を **条件** テーブルで選択します。
ロジック、**演算子**、および **評価結果** 列にリスト矢印が表示されます。いずれかの列の値を編集するには、編集する項目をリストから選択します。**値 1** および **値 2** の各列で、選択した条件の列のセルをダブルクリックして、**条件の選択** ダイアログ ボックスを開きます。このダイアログ ボックスで、いずれかの値を編集できます。
- 複合条件の順序を編集するには、順序を編集する条件を選択してから **上へ移動** または **下へ移動** をクリックします。
論理演算子 (AND、OR) を含む条件も **条件** テーブルの先頭行に移動できます。ただし、論理演算子は条件と一緒に移動されません。条件ロジックの構造を維持するために、演算子は 2 行めに残ります。
- OK** をクリックします。

条件の値の設定

ビジュアル テストに条件ロジックを追加するとき、**条件デザイナ** で条件に対する値を選択したり、リテラル値を作成するには、**条件の選択** ダイアログ ボックスを使用します。

選択 (リテラル)：このオプションを選択すると、以下のリテラル オプションが表示されます。

- リテラル データ**：テキスト文字列、数値、またはブール (true または false) 値のいずれかを入力し、**OK** をクリックします。

1. **条件デザイナ**の**条件の定義**で、変更する値テキストボックスの右側にある...をクリックします。
条件の選択ダイアログボックスが開きます。
2. 値の種類を選択します。
 - コントロールのプロパティを選択する場合は、**プロパティ**をクリックします。
 - 変数を選択する場合は、**変数**をクリックします。
 - アクティブ データ資産を選択する場合は、**アクティブ データ**をクリックします。
 - 新しいリテラル値（文字列、数値、ブール値）を作成する場合は、**リテラル**をクリックします。
 - 指定したコントロールが存在するかどうかを検証する場合は、**存在確認**をクリックします。
3. **プロパティ**を選択した場合は、以下の操作を実行します。
 - a) **コントロールの識別**メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション**：テスト アプリケーションから可視のリスト ボックスまたはコンボ ボックス コントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接リスト ボックスまたはコンボ ボックス コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**：リスト ボックスまたはコンボ ボックス コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
 - b) コントロールの識別後、**プロパティ** リストから目的のプロパティを選択します。
4. **変数**を選択した場合は、以下の操作を実行します。
 - a) **変数の種類** リストから変数の種類を選択するか、**新規作成** をクリックして新しい変数を作成します。利用可能な変数の種類は次のとおりです。
 - **ローカル**：ローカル変数を使用する場合はこれを選択します。ローカル変数は、単一のビジュアル テスト内で定義され使用される変数で、通常、他の種類の変数の値（グローバル変数や入力変数など）を格納します。
 - **入力パラメータ**：ビジュアル テストに渡されるパラメータを使用する場合はこれを選択します。
 - **出力パラメータ**：ビジュアル テストから返されるパラメータを使用する場合はこれを選択します。
 - **予約**：ビジュアル テストの再生や結果に関する情報を保存する、あらかじめ定義された変数を使用する場合はこれを選択します。
 - b) **変数** リストから目的の変数を選択します。
5. **アクティブ データ**を選択した場合は、以下の操作を実行します。
 - a) **アクティブ データ資産** テキスト ボックスの右側にある **選択** メニューを展開します。
 - b) 新しいアクティブ データ資産を作成するには、**アクティブ データの新規作成** をクリックします。詳細については、「アクティブ データ テスト用の新しいデータ ファイルの作成」を参照してください。
 - c) 既存のアクティブ データ資産に関連付けるには、**既存のアクティブ データとの関連付け** をクリックします。詳細については、「アクティブ データ ファイルのデータのビジュアル テスト データへのマップ」を参照してください。
 - d) アクティブ データ資産を指定した後、**列** リストから使用する列を選択します。
6. **リテラル**を選択した場合は、**リテラル データ** テキスト ボックスに文字列、数値、またはブール値を入力します。
7. **存在確認**を選択した場合は、以下の操作を実行します。
 - a) **コントロールの識別**メニューを展開し、次のいずれかを選択します。
 - **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。

- **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケータを編集することもできます。
- b) コントロールが存在することを確認する場合は、**コントロールが存在する** を選択します。
- c) コントロールが存在しないことを確認する場合は、**コントロールが存在しない** を選択します。
- d) **タイムアウト** フィールドには、コントロールが表示される、または表示されなくなるまで Silk Test Workbench が待機する時間を指定します。
- デフォルトのタイムアウトは 0 で、コントロールが表示される、または表示されなくなるまで Silk Test Workbench は待機しないことを意味します。

8. **OK** をクリックします。

式デザイナー

式デザイナー を使用すると、ビジュアル テストの機能を拡張する式をすばやく作成できます。式は、ローカル変数、予約された変数、アクティブ データ資産、演算子、および関数（ローカル変数またはアクティブ データ資産に文字列または数値を返す）の組み合わせです。

たとえば、10 個の異なる注文番号を必要とするアプリケーションをテストする場合、変数の値を増分し、一意の注文番号を返す式を作成します。

```
OrderNumVar=OrderNumVar+1
```

次に、この式を繰り返しループで使用して、ループを複数回実行します。ループが繰り返されるたびに、注文番号は 1 ずつ増分し、ローカル変数 OrderNumVar に格納されている一意の注文番号が返されます。

また、式にアクティブ データ フィールドを含めることもできます。その場合、式にアクティブ データ列を指定し、その列のすべての値を使用して一連のステップを繰り返す繰り返しロジックを設定します。たとえば、アクティブ データ列の値のうち、指定した値（以下の ADDRESS 列に示されている値など）だけを返す式を作成できます。

ADDRESS
Anytown, USA 12345
Anycity, USA 23456
Anyplace, USA 34567
Anywhere, USA 45678

ADDRESS 列の郵便番号のみを返すには、「ADDRESS」というアクティブ データ列、および文字列の右側から指定した数の文字を返す文字列関数 Right を含む式を作成します。

```
localVar = Right (ActiveDataAssetName("ADDRESS"),5)
```

この式では、文字列の右側から 5 文字が返され、戻り値が localVar に格納されます（この例では、「12345」）。アクティブ データ ファイルの行に対して反復処理を行い、上記の列の各郵便番号を返すには、列の各値を使用して式ステップを繰り返す繰り返しロジックを設定します。

式デザイナー では、文字列関数 Right だけでなく、その他の関数と演算も多数サポートされています。

式デザイナーを使用した式の定義

式デザイナー ダイアログ ボックスを使用して、ビジュアル テストの機能を拡張する式を作成または編集できます。式は、ローカル変数、予約された変数、アクティブ データ資産、演算子、および文字列、数値、またはオブジェクトを返す関数を組み合わせて作成します。詳細については、「式デザイナー」を参照してください。


1. ビジュアル テストを開きます。

2. **テストステップ** ペインで、式を挿入する場所の直前のテストステップを選択します。

3. メニューバーから、**挿入 > 式** を選択します。**式デザイナー** が開きます。

4. **出力** リストで、式の結果を格納するローカル変数またはアクティブ データ資産を選択します。

ビジュアル テストに関連付けられたローカル変数とアクティブ データ資産のみがリストされます。新しく作成した変数とアクティブ データ資産は、ビジュアル テストに自動的に関連付けられます。

 **注:** 式を **プロパティ** ペインのプロパティの値として作成する場合は、**出力** リストは表示されません。

5. 省略可能：既存のアクティブ データ資産を選択したり、新しい変数またはアクティブ データ資産を作成したりするには、**出力** リストの隣にある **選択** をクリックします。

以下のいずれか 1 つを選択できます。

- **ローカル変数の新規作成** を選択すると、**ローカル変数の追加** ダイアログ ボックスを使用して、ビジュアル テスト用のローカル変数を作成できます。作成したローカル変数はすべて、**出力** リストに表示され、式の結果を格納するために使用できます。
- **アクティブ データの新規作成** を選択すると、新しいアクティブ データ資産を作成して、ビジュアル テストに関連付けることができます。
- **既存のアクティブ データとの関連付け** を選択すると、**アクティブ データの参照** ダイアログ ボックスを使用して、既存のアクティブ データ資産をビジュアル テストに関連付けることができます。

出力 リストには、関連付けられたアクティブ データ資産の一部であるアクティブ データ ファイルの列名も表示され、式の結果を格納するために使用できます。

6. **エディタ** ウィンドウを使用して、式を作成および編集します。

式には、リテラル値、変数、アクティブ データ列、演算子、および関数を含めることができます。

- a) 式に変数をすばやく追加するには、**変数の選択 (ローカルまたは予約)** リストから目的の変数を選択し、**追加** ボタンをクリックします。
- b) 省略可能：**新規作成** をクリックすると、**ローカル変数の追加** ダイアログ ボックスが開きます。このダイアログ ボックスでは、新しい変数を作成して、それをビジュアル テストに自動的に関連付けることができます。
- c) アクティブ データ フィールドを追加するには、**アクティブ データ フィールドの選択** リストから目的のアクティブ データ フィールドを選択し、**として追加** ボタンをクリックしてデータ型を選択します。

エディタ ウィンドウでのアクティブ データ フィールドの構文は、以下のとおりです。

```
[プロジェクト名].[アクティブ データ名].データ型("列名")
```

サポートされているデータ型は、以下のとおりです。

- | | |
|-----------------------|--|
| テキスト | 選択された アクティブ データ 列のデータは、式内でテキスト データとして使用されます (String データ型)。 |
| 数値 (Long) | 選択された アクティブ データ 列のデータは、式内で数値データとして使用されません (Long データ型)。 |
| 数値 (Double) | 選択された アクティブ データ 列のデータは、式内で数値データとして使用されません (Double データ型)。 |
| ブール値 | 選択された アクティブ データ 列のデータは、式内で True または False として評価されます。(Boolean データ型)。 |
| 数値 (Long Long) | 選択された アクティブ データ 列のデータは、式内で 64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) として使用されます。 |
| 列挙 | 選択された アクティブ データ 列のデータは、式内で一連の値として使用され、1 から n まで順番に番号が付けられます。このデータ型は、パラメーターとして列挙が |


必要な一部のユーザー インターフェイスで使用されます。たとえば、SAP SendVKey は送信するキーとして整数の列挙値を想定しています。限られた個数の一意の値のみを式に保持する場合は、列挙型を宣言します。ビジュアル テストで使用するためのカスタム列挙を作成することはできません。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

- d) 新しいアクティブ データ資産を作成したり、既存のアクティブ データ資産を選択したりするには、**選択** をクリックして目的のオプションを選択します。新しいアクティブ データ資産は、ビジュアル テストに自動的に関連付けられます。
7. **テスト** をクリックして、式の有効性を評価します。テストの結果が **テスト結果** ウィンドウに表示されます。
8. 省略可能 : **エディタ** ウィンドウの内容全体を消去するには、**クリア** をクリックします。
9. **OK** をクリックします。

式デザイナーを使用した式の編集

式デザイナー ダイアログ ボックスを使用して、ビジュアル テストの機能を拡張する式を作成または編集できます。式は、ローカル変数、予約された変数、アクティブ データ資産、演算子、および文字列、数値、またはオブジェクトを返す関数を組み合わせて作成します。詳細については、「式デザイナー」を参照してください。

1. 変更する式が含まれているビジュアル テストを開きます。
 2. **テスト ステップ** ペインで式ステップを選択します。
 3. **プロパティ** ペインで **式** プロパティを選択します。
 4. **式デザイナー** をクリックします。**式デザイナー** が開きます。
 5. **出力** リストで、式の結果を格納するローカル変数またはアクティブ データ資産を選択します。
ビジュアル テストに関連付けられたローカル変数とアクティブ データ資産のみがリストされます。新しく作成した変数とアクティブ データ資産は、ビジュアル テストに自動的に関連付けられます。
-  **注:** 式を **プロパティ** ペインのプロパティの値として作成する場合は、**出力** リストは表示されません。
6. 省略可能 : 既存のアクティブ データ資産を選択したり、新しい変数またはアクティブ データ資産を作成したりするには、**出力** リストの隣にある **選択** をクリックします。
以下のいずれか 1 つを選択できます。

- **ローカル変数の新規作成** を選択すると、**ローカル変数の追加** ダイアログ ボックスを使用して、ビジュアル テスト用のローカル変数を作成できます。作成したローカル変数はすべて、**出力** リストに表示され、式の結果を格納するために使用できます。
- **アクティブ データの新規作成** を選択すると、新しいアクティブ データ資産を作成して、ビジュアル テストに関連付けることができます。
- **既存のアクティブ データとの関連付け** を選択すると、**アクティブ データの参照** ダイアログ ボックスを使用して、既存のアクティブ データ資産をビジュアル テストに関連付けることができます。

出力 リストには、関連付けられたアクティブ データ資産の一部であるアクティブ データ ファイルの列名も表示され、式の結果を格納するために使用できます。

7. **エディタ** ウィンドウを使用して、式を作成および編集します。
式には、リテラル値、変数、アクティブ データ列、演算子、および関数を含めることができます。
 - a) 式に変数をすばやく追加するには、**変数の選択 (ローカルまたは予約)** リストから目的の変数を選択し、**追加** ボタンをクリックします。

- b) 省略可能 : **新規作成** をクリックすると、**ローカル変数の追加** ダイアログ ボックスが開きます。このダイアログ ボックスでは、新しい変数を作成して、それをビジュアル テストに自動的に関連付けることができます。
- c) アクティブ データ フィールドを追加するには、**アクティブ データ フィールドの選択** リストから目的のアクティブ データ フィールドを選択し、**として追加** ボタンをクリックしてデータ型を選択します。

エディタ ウィンドウでのアクティブ データ フィールドの構文は、以下のとおりです。

[プロジェクト名].[アクティブ データ名].データ型("列名")

サポートされているデータ型は、以下のとおりです。

テキスト 選択された **アクティブ データ** 列のデータは、式内でテキスト データとして使用されます (String データ型)。

数値 (Long) 選択された **アクティブ データ** 列のデータは、式内で数値データとして使用されず (Long データ型)。

数値 (Double) 選択された **アクティブ データ** 列のデータは、式内で数値データとして使用されず (Double データ型)。

ブール値 選択された **アクティブ データ** 列のデータは、式内で True または False として評価されます。(Boolean データ型)。

数値 (Long Long) 選択された **アクティブ データ** 列のデータは、式内で 64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) として使用されず。

列挙 選択された **アクティブ データ** 列のデータは、式内で一連の値として使用され、1 から n まで順番に番号が付けられます。このデータ型は、パラメーターとして列挙が必要な一部のユーザー インターフェイスで使用されます。たとえば、SAP SendVKey は送信するキーとして整数の列挙値を想定しています。限られた個数の一意の値のみを式に保持する場合は、列挙型を宣言します。ビジュアル テストで使用するためのカスタム列挙を作成することはできません。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

- d) 新しいアクティブ データ資産を作成したり、既存のアクティブ データ資産を選択したりするには、**選択** をクリックして目的のオプションを選択します。新しいアクティブ データ資産は、ビジュアル テストに自動的に関連付けられます。

8. **テスト** をクリックして、式の有効性を評価します。テストの結果が **テスト結果** ウィンドウに表示されます。

9. 省略可能 : **エディタ** ウィンドウの内容全体を消去するには、**クリア** をクリックします。

100K をクリックします。

式デザイナの演算子の使用

式デザイナ では、以下の演算子がサポートされています。

算術演算子

演算子	演算子名	構文	例
+	プラス演算子	+a	+1

演算子	演算子名	構文	例
-	マイナス演算子 (単項)	-a	-1
+	加算演算子	a + b	1 + 2 = 3
-	減算演算子	a - b	2 - 1 = 1
*	乗算演算子	a * b	2 * 2 = 4
/	除算演算子	a / b	4 / 2 = 2
%	剰余演算子	a % b	5%2=1 (余り)

比較演算子

演算子	演算子名	構文	例
!	ブール論理否定演算子 (単項)	!a	!a (a ではない)
&&	論理 AND	a && b	1 < 5 && 5 < 10 = true
	論理 OR	a b	1 < 5 5 > 10 = true
=	代入演算子	a = b	a = 3
==	Equal To (等値) 演算子	a == b	3 == 3
!=	Not Equal To (非等値) 演算子	a != b	4 != 3
>	Greater Than (より大きい) 演算子	a > b	4 > 3
>=	Greater Than Or Equal To (以上) 演算子	a >= b	4 >= 3
<	Less Than (より小さい) 演算子	a < b	3 < 4
<=	Less Than Or Equal To (以下) 演算子	a <= b	3 <= 4

ビット処理演算子

演算子	演算子名	構文	例
~	ビットごとの NOT (バイナリ数値のビットを反転することで、式で論理否定を実行)	a = ~ b	~1 = -2、ここで 00000001 = 1 11111110 = -2
&	ビットごとの AND (2つの式で論理積を実行)	a = b & c	1 & 1 = True 0 & 1 = False 1 & 0 = False 0 & 0 = False
	ビットごとの OR (2つの式で論理和を実行)	a = b c	1 1 = True 0 1 = True 1 0 = True

演算子	演算子名	構文	例
^	ビットごとのXOR (2つの式で排他的論理和を実行)	$a = b \wedge c$	$0 \mid 0 = \text{False}$
			$1 \wedge 1 = \text{False}$
			$0 \wedge 1 = \text{True}$
			$1 \wedge 0 = \text{True}$
			$0 \wedge 0 = \text{False}$

式デザイナーの関数の使用

式デザイナーでは、以下の関数がサポートされています。

数学関数


関数/説明	構文	例
Abs: 数値の絶対値を返します。	Abs(式)	Abs(-2) と Abs(2) は両方とも 2 を返します。
Round: 最も近い整数に丸めた数値を返します。	Round(式)	Round(1.3333) は 1 を返します。
Max: 2 つの数値を比較し、大きい方の数値を返します。	Max(数値 1, 数値 2)	Max(-123,12) は 12 を返します。
Min: 2 つの数値を比較し、小さい方の数値を返します。	Min(数値 1, 数値 2)	Min(-123,12) は -123 を返します。

文字列関数

関数/説明	構文	例
Contains: 文字列 2 が文字列 1 に含まれているかどうかを返します。	Contains(文字列 1, 文字列 2)	Contains("Hello World", "Hello") は true を返します。 Contains("Hello World", "Goodbye") は false を返します。
Find: 文字列内の特定の文字列の最初の出現位置を返します。	Find(文字列 1, 文字列 2)	Find("abcd","c") は 3 を返します。
FormatDate: 日付、日付形式、およびロケールに基づく日付文字列を返します。	FormatDate(日付, 日付形式, ロケール)	FormatDate("May 13, 2009", "MM/dd/yy", "en-US") は「05/13/09」を返します。
FormatTime: 時刻、時刻形式、およびロケールに基づく時刻文字列を返します。	FormatTime(時刻, 時刻形式, ロケール)	FormatTime("23:02:33", "hh:mm tt", "en-US") は「11:02 PM」を返します。
Left: 文字列の左側から指定した数の文字を返します。	Left(文字列, 長さ)	Left("abcd",2) は「ab」を返します。
Length: 文字列の文字数を返します。	Length(文字列)	Length("abcd") は 4 を返します。
MakeLower: テキスト文字列を小文字に変換します。	Makelower(文字列)	Makelower("ABCD") は「abcd」を返します。

関数/説明	構文	例
ToUpper : テキスト文字列を大文字に変換します。	ToUpper(文字列)	ToUpper("abcd") は「ABCD」を返します。
Mid : 文字列の中央から指定した数の文字を返します。	Mid(文字列, 開始位置, 長さ)	Mid("abcd",2,2) は「bc」を返します。
Replace : 指定した回数だけ、指定した部分文字列が別の部分文字列に置換されたテキスト文字列を返します。	Replace(文字列, 置換対象の部分文字列, 置換後の部分文字列)	Replace("abc def","bc","xy") は「axy def」を返します。
Right : 文字列の右側から指定した数の文字を返します。	Right(文字列, 長さ)	Right("abcd",2) は「cd」を返します。
Trim : 指定した文字列を、先頭と末尾のスペースを削除して返します。	Trim(文字列)	Trim(" abcd ") は「abcd」を返します。
TrimLeft : 指定した文字列を、先頭のスペースを削除して返します。	TrimLeft(文字列)	TrimLeft(" abcd ") は「abcd」を返します。
TrimRight : 指定した文字列を、末尾のスペースを削除して返します。	TrimRight(文字列)	TrimRight(" abcd ") は「abcd」を返します。

変換関数

関数/戻り値の型	構文	式の範囲
ToBool : Boolean	ToBool(式)	有効な文字列または数値の式
ToDouble : Double	ToDouble(式)	1.7E +/- 308
ToLong : Long	ToLong(式)	-2,147,483,648 から 2,147,483,647 (端数は丸められる)
ToString : Text	ToString(式)	ToString の戻り値は式の内容によって異なります。
ToLongLong : Integer	ToLongLong	-9,223,372,036,854,775,808 から 9,223,372,036,854,775,807
ToEnumeration : Integer	ToEnumeration	-9,223,372,036,854,775,808 から 9,223,372,036,854,775,807  注: Open Agent では 64 ビットのデータ型がサポートされないため、この式に推奨される範囲は -2,147,483,648 から 2,147,483,647 までです。

変換規則

データ型を変換する場合は、以下の規則に従ってください。

Double 型から Long 型への変換

小数点は切り捨てられます。Long 型の範囲外の数値はラップされます。たとえば、Double 型の数値 49.6 は 49 に丸められます (つまり、.6 は切り捨てられます)。Double 型の数値 -2147483649 は Long 型の上限 (+214748367) にラップされます。

String 型から Double 型または Long 型への変換 変換は、認識不可の最初の文字（英字など）で停止します。Long 型の場合、小数点は切り捨てられ、Long 型の範囲外の数値はラップされます。たとえば、「123.7 oranges」は、Double 型の「123.7」および Long 型の「123」に変換されます。

Boolean 型への変換 ゼロ以外の数値、テキスト「true」、または数値に変換されるすべてのテキストは、True に変換されます。ゼロ、テキスト「false」、または数値に変換できないすべてのテキストは、False に変換されます。

Boolean 型から Double 型または Long 型への変換 True = 1 および False = 0。

FormatDate 関数

説明

ユーザーが指定した日付、日付形式、およびロケールに基づいて文字列式を返す式デザイナの関数。

構文

```
local variable = FormatDate (Date, Format, Locale)
```

FormatDate 関数の構文のパラメータは以下のとおりです。

Date 指定されたロケールに有効な日付文字列または日付文字列が含まれる変数。現在のシステムの日付を指定するには、「Current」を使用します。日付文字列を指定するには、希望する日付文字列を引用符で囲んで入力します（例："January 5th, 2009"）。日付文字列が含まれる変数の場合は、引用符なしで変数名を使用します。

Format 指定したロケールに対して有効な以下の要素の組み合わせで構成される日付形式。

- 日**
- d : 日を数値で表し、数値が 1 桁の場合は先頭にゼロを付けません（「1」から「9」、
「10」から「31」）。
 - dd : 日を数値で表し、数値が 1 桁の場合は先頭にゼロを付けます（「01」から「09」、
「10」から「31」）。
 - ddd : 曜日を表す 3 文字の略語（英語の Mon、Tue、Wed など）。
 - dddd : 曜日を略さずに表します（英語の Monday、Tuesday、Wednesday など）。
- 月**
- M : 月を数値で表し、数値が 1 桁の場合は先頭にゼロを付けません（「1」から「9」、
「10」から「12」）。
 - MM : 月を数値で表し、数値が 1 桁の場合は先頭にゼロを付けます（「01」から
「09」、
「10」から「12」）。
 - MMM : 月を表す 3 文字の略語（英語の Jan、Feb、Oct など）。
 - MMMM : 月を略さずに表します。
- 年**
- y : 年の最後の 2 桁の数値で表し、10 年未満である場合は先頭にゼロを付けません
（「0」から「9」、
「10」から「99」）。
 - yy : 年の最後の 2 桁を数値で表し、10 年未満である場合は先頭にゼロを付けます（
「00」
から「09」、
「10」から「99」）。
 - yyy : 年の最後の 2 桁を数値で表し、10 年未満である場合は先頭にゼロを付けます
（
「00」
から「09」、
「10」から「99」）。
 - yyyy : 年を略さずに 4 桁の数値で表します。

元号 gg : 元号。



注: Silk Test Workbench では、指定された日付とロケールに元号が存在するかどうかを検証されます。存在する場合は、その元号が Silk Test Workbench によってデフォルトのカレンダーに使用されます。存在しない場合は、同じロケールの別のカレンダーに元号が存在するかどうか Silk Test Workbench によって判断されます。別のカレンダーが存在する場合は、すべての文字列の形式が Silk Test Workbench によってそのカレンダーに合わせて設定されます。存在しない場合、文字列の形式は Silk Test Workbench によってデフォルトのカレンダーに合わせて設定され、元号形式 (gg) は無視されます。

Locale 一般的に、ロケールでは、ユーザーが自分のユーザー インターフェイスに表示するためのユーザーの言語や国、日付と時刻に使用する形式などの任意の特別なバリエーションを設定を定義します。

Locale パラメータにより、FormatDate 関数の出力で日付文字列が解釈され、表示される方法に対して、指定したロケール定義が適用されます。このパラメータの値は、Windows オペレーティングシステムのロケール名に基づきます。たとえば、フランス語 (カナダ) のロケール名は「fr-CA」です。サポートされているロケールの一覧については、Microsoft の『National Language Support Reference』の「Locale Identifier Constants and Strings」というトピックを参照してください。



注: 使用可能なロケールは、オペレーティング システムのバージョンと[地域のオプション]の設定に基づきます。

解説

すべてのパラメータが必須です。Format または Locale パラメータのデフォルト値を使用するには、式に「Default」と入力します。以下の例では、FormatDate 関数により、現在のシステムの日付が、オペレーティング システムの日付形式とロケールを使用して返されます。

```
FormatDate("Current", "Default", "Default")
```

以下の各例は、Format パラメータの要素を使用して FormatDate 関数の出力をカスタマイズする方法を示しています。

```
FormatDate("Jan 9, 2009", "dddd, MMM-dd-yyyy",  
"Default") returns "Friday, Jan-09-2009"
```

```
FormatDate("Jan 9, 2009", "d/M/yy", "Default")  
returns "9/1/09"
```

```
FormatDate("Jan 9, 2009", "yy", "Default")  
returns "09"
```

以下の例は英語のオペレーティング システムで実行するもので、Locale パラメータを使用して、フランス語のロケール日付をスペイン語のロケール日付に変換する方法を示しています。この処理は 2 段階で行われ、最初に、数値ではないフランス語の日付 (2009 janvier 9) を、フランス語の数値の日付 (2009-01-09) 形式に変換します。次に、数値の日付をスペイン語の日付 (2009-enero-09) に変換します。

```
FormatDate("2009 janvier 9", "yyyy-MM-dd",  
"fr-FR") returns "2009-01-09"
```

```
FormatDate(FrNumericDate, "yyyy-MMMM-dd", "es-ES")  
returns "2009-enero-09"
```


FormatTime 関数

説明

ユーザーが指定した時刻、時刻形式、およびロケールに基づいて文字列式を返す式デザイナーの関数。

構文

```
local variable = FormatTime(Time, Format, Locale)
```

FormatTime 関数の構文のパラメータは以下のとおりです。

Time 指定されたロケールに有効な時刻文字列または時刻文字列が含まれる変数。現在のシステムの時刻を指定するには、「Current」を使用します。時刻文字列を指定するには、希望する時刻文字列を引用符で囲んで入力します（例："01:12:23"）。時刻文字列が含まれる変数の場合は、引用符なしで変数名を使用します。

Format 指定したロケールに対して有効な以下の要素の組み合わせで構成される時刻形式。

時間

- h：時間を 12 時間形式で表し、時間が 1 桁の場合は先頭にゼロを付けません。
- H：時間を 24 時間形式で表し、時間が 1 桁の場合は先頭にゼロを付けません。
- hh：時間を 12 時間形式で表し、時間が 1 桁の場合は先頭にゼロを付けます。
- HH：時間を 24 時間形式で表し、時間が 1 桁の場合は先頭にゼロを付けます。

分

- m：分を表し、分が 1 桁の場合は先頭にゼロを付けません。
- mm：分を表し、分が 1 桁の場合は先頭にゼロを付けます。

秒

- s：秒を表し、秒が 1 桁の場合は先頭にゼロを付けません。
- ss：秒を表し、秒が 1 桁の場合は先頭にゼロを付けます。

時刻マーカー

- t：1 文字の時刻マーカー文字列（A または P など）。
- tt：午前と午後を複数文字の文字列で表します（AM、PM など）。

Locale 一般的に、Locale には、ユーザーが自分のユーザー インターフェイスに表示するためのユーザーの言語や国、日付と時刻に使用する形式などの、任意の特別なバリエーションを設定を定義します。

Locale パラメータにより、FormatTime 関数の出力で時刻文字列が解釈され、表示される方法に対して、指定したロケール定義が適用されます。このパラメータの値は、Windows オペレーティングシステムのロケール名に基づきます。たとえば、フランス語（カナダ）のロケール名は「fr-CA」です。サポートされているロケールの一覧については、Microsoft の『National Language Support Reference』の「Locale Identifier Constants and Strings」というトピックを参照してください。



注: 使用可能なロケールは、オペレーティング システムのバージョンと[地域のオプション]の設定に基づきます。

解説

すべてのパラメータが必須です。Format または Locale パラメータのデフォルト値を使用するには、式に「Default」と入力します。以下の例では、FormatTime 関数により、現在のシステムの時刻が、オペレーティングシステムの時刻形式とロケールを使用して返されます。

```
FormatTime ("Current","Default", "Default")
```

以下の各例は、Format パラメータの要素を使用して FormatTime 関数の出力をカスタマイズする方法を示しています。

```
FormatTime("9:05:08 PM", "hh:mm:ss tt", "en-US") returns "09:05:08 PM"
```

```
FormatTime("9:05:08 PM", "HH:mm:ss", "en-US") returns "21:05:08"
```

```
FormatTime("9:05:08 PM", "tt", "Default") returns "PM"
```

以下の例は、Locale パラメータを使用して、英語（米国）と英語（英国）の各ロケールの出力に影響を与える方法を示しています。

```
FormatTime("14:30", "Default", "en-US") returns "2:30:00 PM"
```

```
FormatTime("14:30", "Default", "en-GB") returns "14:30:00"
```

自動ステップの操作

このセクションでは、ビジュアルテストで自動ステップを使用する方法を説明するトピックを示します。

記録されない自動テストステップの作成

自動テストステップとは、アプリケーションをテストするために、テストアプリケーションのコントロールに対して操作を実行するステップのことです。たとえば、テストアプリケーションで項目の選択、テキストの入力、コントロールのクリックなどを行うステップが自動ステップです。

自動ステップは、通常は記録します。ビジュアルテストの記録中にテストアプリケーションのコントロールを操作すると、Silk Test Workbench によって、通常は個々の操作（クリックなど）が単独のステップとして記録されます。

ただし、記録を行わずに自動テストステップを作成して、ビジュアルテストにそのステップを挿入することもできます。記録を行わずに自動ステップを作成すると、テスト計画にわずかな変更が生じた場合に、テスト全体を記録し直すことなくその変更を吸収できます。自動テストステップを作成するには、**オブジェクトの識別** ダイアログ ボックスを使用してテストアプリケーションを操作するか、**画面プレビュー** でキャプチャされた画面イメージを操作します。**オブジェクトの識別** ダイアログ ボックスを使用すると、Silk Test Workbench によってテストアプリケーションのさまざまなコントロールがどのように認識されるかを確認できるため、アプリケーションのテストを最適化する方法を理解しやすくなります。

テストアプリケーションを使用したコントロールの自動ステップの作成

テストアプリケーションを操作して、テスト操作を実行する個々のステップを作成できます。個々のステップを作成するということは、ビジュアルテストを記録し直す必要がなくなるということです。必要に応じて、テストアプリケーションに対して追加の操作を作成するだけで済みます。

画面プレビュー 内にキャプチャされた画面を使用して、コントロールに対して追加操作を作成することもできます。

1. コントロールを識別するビジュアルテストを開きます。
2. コントロールが含まれるテストアプリケーションを開きます。
3. 挿入ポイントの直前のステップを選択します。
選択したステップで、コントロールに対して操作を実行します。
4. **挿入 > コントロール > テスト対象アプリケーションから選択** を選択します。**記録** ダイアログ ボックスが開きます。
5. テストアプリケーションで使用するコントロールを選択します。**操作の選択** ダイアログ ボックスが開きます。
6. コントロールに対する操作を指定します。選択したステップの下に自動ステップが表示されます。ステップのプロパティが**プロパティ** ペインに表示されます。選択した操作は、**操作** カテゴリに**メソッド** プロパティとして表示されます。ステップのプロパティを編集すると、操作の実行方法をさらに制御できます。

画面プレビューを使用したコントロールの自動ステップの作成

画面プレビューを使用してコントロールの自動ステップを作成する前に、オプション **コントロールのキャプチャ** がはいに設定されていることを確認してください。

テストアプリケーションを操作して、テスト操作を実行する個々のステップを作成できます。個々のステップを作成するということは、ビジュアルテストを記録し直す必要がなくなるということです。必要に応じて、**画面プレビュー** でキャプチャされているテストアプリケーションの画面に対して追加操作を作成します。

テストアプリケーションを使用して、コントロールに追加操作を作成することもできます。

1. コントロールの識別元となる、キャプチャされた画面が含まれるビジュアルテストを開きます。
2. **画面プレビュー** にテストアプリケーションの画面またはコントロールを表示するステップを選択します。これを行うには、[テストステップ] ペインでステップをクリックするか、ストーリーボードでサムネイルをクリックします。
画面プレビュー で個々のコントロールを見やすくするには、**画面プレビュー** の **操作 > ズーム** を選択し、ズームの割合を選択します。
3. **画面プレビュー** で目的のコントロールを表示し、**操作 > コントロールの挿入 > 画面プレビューから選択** を選択します。
4. **画面プレビュー** でコントロールを識別します。**操作の選択** ダイアログボックスが開きます。
5. コントロールに対する操作を指定します。選択したステップの下に自動ステップが表示されます。ステップのプロパティが**プロパティ** ペインに表示されます。選択した操作は、**操作** カテゴリに**メソッド** プロパティとして表示されます。ステップのプロパティを編集すると、操作の実行方法をさらに制御できます。
6. **OK** をクリックします。

オブジェクトの識別ダイアログを使用したコントロールの自動ステップの作成

テストアプリケーションを操作して、テスト操作を実行する個々のステップを作成できます。個々のステップを作成するということは、ビジュアルテストを記録し直す必要がなくなるということです。必要に応じて、テストアプリケーションに対して追加の操作を作成するだけで済みます。

オブジェクトの識別 ダイアログボックスを使用して、非表示コントロールに対する追加の操作を作成することもできます。

1. コントロールを識別するビジュアルテストを開きます。
2. コントロールが含まれるテストアプリケーションを開きます。
3. 挿入ポイントの直前のステップを選択します。
選択したステップで、コントロールに対して操作を実行します。
4. **挿入 > コントロール > オブジェクトの識別ダイアログ** を選択します。
5. Web アプリケーションをテストする場合は、**ブラウザーの選択** ダイアログボックスが開きます。コントロールを識別するブラウザーを選択し、**識別の開始** をクリックします。**オブジェクトの識別** ダイアログボックスが開きます。
6. **識別の開始** をクリックします。
7. テストしているアプリケーションにカーソルを移動します。カーソルをアプリケーション内の各コントロール上に動かすたびにコントロールが強調表示されます。関連するロケータ文字列またはオブジェクトマップ項目は、**選択済みロケータ** テキストボックスに表示されます。
8. 使用するコントロールが強調表示されているときに、コントロールをクリックします。
9. 省略可能：**ロケータの詳細の表示** をクリックすると、ロケータ ツリーや関係する属性のすべてが**ロケータ属性** テーブルに表示されます。
10. 省略可能：記録したロケータ属性は、**ロケータ属性** テーブルの別のロケータ属性で置き換えることができます。

たとえば、記録したロケータは以下のように表示されます。

```
/BrowserApplication//BrowserWindow//input[@id='loginButton']
```

ロケーター属性 テーブルに textContents Login がリストされている場合、以下のようにしてロケーターを手動で変更できます。

```
/BrowserApplication//BrowserWindow//input[@textContents='Login']
```

新しいロケーターは、**選択済みロケーター** テキスト ボックスに表示されます。

11テスト をクリックして、ロケーターがコントロールを正しく認識することを確認します。

12ロケーターを使用する新しいステップをビジュアル テストに追加するには、**貼り付け** をクリックします。**操作の選択** ダイアログ ボックスが開きます。コントロールに対して実行する操作を選択し、**OK** をクリックします。

ビジュアル テスト内でのコントロール プロパティ データの使用

Silk Test Workbench では、テスト アプリケーションの任意のコントロールのプロパティ値を、ビジュアル テストのデータとして使用できます。この場合、プロパティ変数を、ビジュアル テスト内で定義されているローカル変数に設定します。たとえば、コントロールの ID を変数に設定してからテスト ロジックでこの変数を使用すると、アプリケーションの他のコントロールで同じ ID が使用されていないことを確認できます。

変数に設定できるプロパティは、そのコントロールで使用可能なプロパティによって異なります。すべてのコントロールで同じプロパティが使用可能なわけではありません。

コントロール プロパティ 値を変数に設定するには、ビジュアル テストにその変数を作成する必要があります。

1. コントロール プロパティを変数に設定するビジュアル テストを開きます。
2. コントロール プロパティの値を保持するローカル変数を作成します。
3. コントロールのプロパティを変数に設定するステップを挿入する場所の直前のステップを選択します。目的のプロパティ データを持つコントロールが含まれるアプリケーションが、ステップを挿入するテスト内のポイントで使用可能になっていることを確認してください。
4. **挿入 > コントロールのプロパティ** を選択します。選択したステップのあとにステップが作成されます。ステップのプロパティが **プロパティ** ペインに表示されます。
5. **プロパティ** ペインの **ロケーター** テキスト ボックスで、変数を割り当てるコントロールを識別します。
 - **テスト対象アプリケーション**： テスト アプリケーションから可視のリスト ボックスまたはコンボ ボックス コントロールを識別するには、このボタンをクリックします。
 - **画面プレビュー**： テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接リスト ボックスまたはコンボ ボックス コントロールを識別するには、このボタンをクリックします。
 - **オブジェクトの識別ダイアログ**： リスト ボックスまたはコンボ ボックス コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケーターを編集することもできます。
 - **選択**： このボタンをクリックして、リテラル、変数、式の結果、またはアクティブ データ値を割り当てます。
6. **プロパティ** リストから、コントロールのプロパティとしてリテラル、変数、式、またはアクティブ データのいずれを使用するかを指定します。
この値はローカル変数に格納されます。
7. **ローカル変数名** リストから、選択したコントロールおよびプロパティで使用する変数を選択します。
8. 省略可能：**その他** カテゴリのプロパティを編集し、ステップに関する一般的な情報を入力します。

ビジュアル テストにおける画面プレビューとステップの同期

Silk Test Workbench では、記録中と再生中にアプリケーションのスナップショットをキャプチャします。このスナップショットには、キャプチャされた画面と、さまざまなプロパティを持つ各種コントロールの列挙が含まれます。キャプチャされた画面は **画面プレビュー** に表示されます。

Silk Test Workbench では、スナップショットは以下のような状況のときに取得されます。

- 記録中のすべての自動テスト ステップの前。



注: SAP アプリケーションの場合、スナップショットはすべての自動テスト ステップの前ではなく、画面が変わったときに発生します。

- ビジュアル テストで Using ステップを実行した結果のスナップショット。
- 再生エラーが発生した場合。



ヒント: アプリケーション画面の内容を更新するのに時間のかかるテスト アプリケーションの場合には、アプリケーション画面のスナップショットを取得する前に待機するよう Silk Test Workbench を設定できます。詳細については、**アタッチとスクリーンショット間の遅延** オプションを参照してください。

自動テスト ステップ

すべての自動テスト ステップには独自のスナップショットがあり、**画面プレビュー** では (グループ画面) または (ステップ画面) というラベルが付けられます。(グループ画面) には、現在のダイアログ ボックスまたは Web ページのメイン コンテキストが含まれています。一方、(ステップ画面) は、現在のダイアログ ボックスまたは Web ページでの特定の操作に関連付けられています。

Silk Test Workbench では、アプリケーションのスナップショットは、そのスナップショットを参照する実際のテスト ステップの前に取られます。画面コンテキストが変わる自動ステップ (新しいダイアログ ボックスまたは Web ページを開くなど) では、Using ステップは、自動ステップの前のアプリケーション状態を表す画面キャプチャに関連付けられます。**画面プレビュー** では、Using ステップには (グループ画面) というラベルが付けられます。たとえば、Web ページの **画面プレビュー** での (グループ画面) は、**テスト ステップ** ペインの Using Browser Application 'BrowserApplication' ステップと対応する場合があります。Using ステップに続くテスト ステップは、現在のダイアログ ボックスまたは Web ページのコンテキストで発生する単一の操作に対応し、**画面プレビュー** では (ステップ画面) というラベルが付けられます。たとえば、テキストフィールドに郵便番号を入力した場合に、対応するテスト ステップは「'92109' を入力します」であるとします。**画面プレビュー** では、入力した値の前の郵便番号のテキストフィールドと郵便番号の値の周囲に黒い四角形が表示されます。したがって、92109 の前の値が 18966 だった場合、**画面プレビュー** には 18966 が表示されます。

Using ステップを無効にした場合、Using ステップに続くステップは、無効にした Using ステップの前の Using ステップに関連付けられます。たとえば、Using ステップ 26 を無効にした場合、それに続くテスト ステップはステップ 26 の前の Using ステップを使用します。



注: 画面プレビューからコントロールを挿入すると、**画面プレビュー** に (グループ画面) スナップショットが表示されます。コントロールは記録されなかったため、スナップショットは取得されませんでした。この結果、(グループ画面) が使用されます。

その他のテスト ステップのタイプ

ほとんどの場合、自動ステップに関連付けられていないテスト ステップでは、**画面プレビュー** にスナップショットは表示されません。たとえば、ビジュアル テストにスクリプトを挿入すると、**画面プレビュー** には **テスト ステップ** ペインに表示されるテキストが表示されます。

設定の構成

デフォルトでは、Silk Test Workbench によって記録中にグループと個別両方の画面キャプチャが記録されます。これらの設定は、**オプション** ダイアログ ボックスを使用して、グループ画面キャプチャのみを記録するように、または画面キャプチャを記録しないように変更できます。**ツール > オプション** を選択し、**記録** を展開します。**出力** をクリックし、**ビジュアル テスト** をクリックして、これらのオプションを構成します。

オプション ダイアログ ボックスでは、ビジュアル テストのグループと個別両方の画面キャプチャの再生画面キャプチャ設定を構成するか、すべてのビジュアル テストに適用されるグローバル オプションを設定できます。**ツール > オプション** を選択し、**再生** を展開します。**結果** をクリックし、**ビジュアル テスト** をクリックして、これらのオプションを構成します。

ビジュアルテストの個別の設定を構成するには、ビジュアルテストを記録し、次に、グローバル再生値を上書きする再生設定の設定ステップを挿入します。設定には、画面キャプチャと画面キャプチャ (テストステップ) があります。

たとえば、パフォーマンスを向上させるために、特定のステップについては **画面キャプチャ (テストステップ)** の再生設定を **いいえ** に設定し、個別の画面キャプチャが必要な残りのステップについては **画面キャプチャ (テストステップ)** を **はい** に戻す場合があります。

画面テストステップのキャプチャ画面の更新

ビジュアルテストまたは結果の **画面プレビュー** から、以前にキャプチャされた画面テストステップの画面を即座に更新することができます。キャプチャされたイメージは、関連するステップが実行される前のアプリケーションの状態を示します。Silk Test Workbench では、キャプチャされた画面を更新する方法には以下の2つがあります。

- | | |
|--------------------------------|--|
| テストアプリケーションからの画面の再キャプチャ | この方法は、ビジュアルテストの 画面プレビュー から実行します。テストアプリケーションを開き、新しい画面が含まれるウィンドウを表示して、それを再キャプチャします。 |
| 再生中にキャプチャされた画面の使用 | この方法は、結果の 画面プレビュー から実行します。テストアプリケーションにアクセスする必要はありません。 |

再生中に記録された画面を使用したキャプチャ画面の更新

ビジュアルテストまたは結果の **画面プレビュー** から、以前にキャプチャされた画面テストステップの画面を即座に更新することができます。キャプチャされたイメージは、関連するステップが実行される前のアプリケーションの状態を示します。

1. ビジュアルテストの結果を開きます。
2. **テストステップ** ペインで、更新が必要な画面が含まれるステップを選択するか、画面が含まれるストーリーボードのサムネイルをクリックします。
3. 現在のステップのみを更新する場合は、**画面プレビュー** または **テストステップ** ペインから **操作 > 画面の更新** を選択します。Silk Test Workbench により、ビジュアルテストが開き、更新された画面が **画面プレビュー** に表示され、ビジュアルテストの画面項目を更新するか尋ねる確認メッセージボックスが表示されます。この機能を利用するには、結果とビジュアルテストの名前を同じにする必要があります。ビジュアルテストの名前を変更した場合は、その結果の名前も変更してください。変更しなかった場合は、ビジュアルテストの画面項目を変更しようとしても、Silk Test Workbench でビジュアルテストが見つかりません。



注: 画面テストステップを更新する前に、記録中にキャプチャされた画面と再生中にキャプチャされた画面の違いを表示することができます。**操作 > 相違点の表示 > 横に並べる** を選択してください。

4. **はい** をクリックして、再生中にキャプチャされた画面でビジュアルテストを更新します。ビジュアルテストが開きます。
5. ビジュアルテストのすべての画面を更新する場合は、**画面プレビュー** または **テストステップ** ペインから **操作 > すべての画面を更新** を選択します。この機能を利用するには、結果とビジュアルテストの名前を同じにする必要があります。ビジュアルテストの名前を変更した場合は、その結果の名前も変更してください。変更しなかった場合は、ビジュアルテストの画面項目を変更しようとしても、Silk Test Workbench でビジュアルテストが見つかりません。
6. 確認メッセージボックスの **結果へ戻る** チェックボックスをオンにすると、**はい** をクリックしたあとで **結果** ウィンドウが表示されます。

テストアプリケーションからキャプチャされた画面の更新

ビジュアルテストまたは結果の **画面プレビュー** から、以前にキャプチャされた画面テストステップの画面を即座に更新することができます。キャプチャされたイメージは、関連するステップが実行される前のアプリケーションの状態を示します。

1. テストアプリケーションを起動します。
2. テストアプリケーションを操作して、ビジュアルテストで情報の更新が必要な画面が含まれるウィンドウを表示します。
3. ビジュアルテストを開き、更新が必要な画面が含まれる画面ステップを選択するか、画面が含まれるストーリーボードのサムネイルをクリックします。
4. **操作 > 画面の更新** を選択します。テストアプリケーションで更新するテストステップに該当する画面が見つかり、Silk Test Workbench によってその画面が強調表示されます。テスト画面の周囲が四角の境界線で囲まれて点滅します。テストアプリケーションで画面が見つからない場合は、テストアプリケーションでウィンドウが見つからないことを示すメッセージが表示されます。ウィンドウの名前が変更されているか、Silk Test Workbench でウィンドウを認識できないようにウィンドウのプロパティが変更されている場合があります。**はい** をクリックしてテストアプリケーションで更新されたウィンドウを再識別するか、別のウィンドウを識別してください。ウィンドウ情報を再キャプチャしないでビジュアルテストに戻るには、**いいえ** をクリックします。
5. テストアプリケーションの画面からキャプチャされた更新情報で、テストステップを更新するかどうかを確認するメッセージが表示されます。
 - テストステップを更新するには、**はい** をクリックします。
 - テストステップを更新しないでメッセージボックスを閉じるには、**いいえ** をクリックします。

ビジュアルテストでの項目識別子の使用

ビジュアルテストでの項目識別子は、単純な数値ではなく、かぎ括弧で囲んだインデックスで指定します。

例

たとえば、GetCellText メソッドで DomTable の 2 行 2 列のセルのテキストを返すには、次のステップをビジュアルテストに追加します。

行 '[2]'、列 '[2]' のセルテキストを返します。また、戻り値を st_LastReturnValue に割り当てます

このステップのプロパティのパラメータセクションで、row パラメータを [2] に、そして、col パラメータも [2] に設定します。



注: ビジュアルテストでは、項目識別子にはかぎ括弧が必要なため、これらのパラメータとして [2] ではなく 2 を設定すると、正しく動作しません。

ビジュアルテストでのテストデータの再利用

ビジュアルテストで使用できる再利用の種類を示します。

ビジュアルテストにおける変数

アプリケーションのテストでは、通常はデータを使用する必要があります。アプリケーションの一部をテストするためにデータを使用したり、そのアプリケーションやその他のアプリケーション全体でデータを使用したりできます。データをフィールドに入力して、そのフィールドがどのようにユーザー入力を受け入れるかをテストできます。また、データを使用して、テストアプリケーションに対してテストがどのように再生されるかを確認することもできます。

記録中、コントロールへのデータ入力はリテラルデータとして記録されます。たとえば、WPF TextBox コントロールに対して入力された値がある場合、その結果のテストステップは次のようになります：「"Ford" を入力します」。

この値は有効ですが、これではビジュアルテストはこのテストステップで Ford 以外の値のデータを使用することができません。ただし、値 Ford を、このデータを表す変数で置き換えると、変数で使用する任意のデータが TextBox コントロールに入力されます。

変数を使用すると、ビジュアルテストで使用するデータが定数である必要がなくなるため、テストの柔軟性が高まります。

ビジュアルテストに変数を定義する場合は、その変数がどのような種類のデータを保持できるかを定義する必要があります。このような変数の特性を、データ型と呼びます。ビジュアルテストでは、以下のデータ型の変数を使用できます。

- テキスト (String)
- 数値 (Long)
- 数値 (Double)
- ブール値 (True/False)
- 数値 (Long Long)
- 列挙

たとえば、Silk Test Workbench でコントロールへのデータ入力が記録される時、記録された値が二重引用符 ("") で囲まれている場合、通常はテキストデータ型であり、テキストデータ型の変数で置き換えることができます。

ビジュアルテストでは、以下のタイプの変数を使用します。

- ローカル：通常 1 つのビジュアルテスト内で定義され、使用される変数です。ローカル変数を使用して、グローバル変数やビジュアルテスト変数などの他のタイプの変数の値を格納することができます。**挿入 > 変数 > ローカルの追加** を選択してローカル変数を追加するか、または <<開始>> ステップの **プロパティ** ペインでローカル変数を定義します。
- グローバル：グローバル変数は、ビジュアルテストのコンテキスト内で定義されますが、そのグローバル変数が含まれているビジュアルテストを実行する、任意のビジュアルテストで使用できます。**ビジュアルナビゲータ** でビジュアルテストを操作している場合は、**挿入 > 変数 > グローバルの設定** を選択して、グローバル変数の値を定義します。他のビジュアルテストでは、**挿入 > 変数 > グローバルの取得** を選択して、グローバル変数の値をローカル変数に割り当てます。
- パラメータ：1 つのビジュアルテストから別のビジュアルテストに渡される変数です。
- 入力/出力パラメータ：入力パラメータを使用して、ビジュアルテストからスクリプトにデータを渡します。出力パラメータを使用して、スクリプトからビジュアルテストにデータを渡します。
- 予約：事前に定義されている変数です。この変数を使用して、ビジュアルテストの再生に関する情報やその他の有用な再生結果情報を取得することができます。予約変数の名前は、必ず接頭部 ST で始まります。予約変数は読み取り専用ですが、式の中で他のタイプの変数と共に使用できます。



ヒント: 変数を使用することによって、スクリプトでデータを再利用することもできます。詳細については、「スクリプトにおける変数」を参照してください。


ビジュアルテストの再生における、変数の使用状況の監視


Silk Test Workbench では、ビジュアルテストの再生時に変数の使用を追跡できます。変数の使用状況と再生中に変数に格納される値を追跡することによって、ビジュアルテストのデバッグに関する有益な情報を入手できます。

ビジュアルテストを再生する前に、ローカル変数ウィンドウを表示します。このウィンドウには、ビジュアルテストの再生中に変数が作成、使用、削除されたときに変数の使用状況が表示されます。

ローカル変数ウィンドウでビジュアルテストに 1 つまたは複数のブレークポイントを設定し、テストステップを段階的に再生して、テスト実行の重要なポイントで変数の使用状況に関するスナップショットを取得できます。

1. 再生中に監視する変数が含まれているビジュアルテストを開きます。
2. **デバッグ > ローカル変数** を選択します。ローカル変数ウィンドウが開きます。このウィンドウは、移動、ドッキング、非表示、またはサイズ変更が可能です。
3. 選択した場所で再生を停止するようにブレークポイントを設定し、変数を追跡しやすくします。
4. ビジュアルテストを再生します。デバッグモードの場合は選択した場所で再生が停止し、既存の変数ごとにその現在値がローカル変数ウィンドウに表示されます。

 **ヒント:** また、ローカル変数 ウィンドウには、変数を使用しているビジュアル テストの名前も表示されます。これは、複数のビジュアル テストを再生している場合、またはビジュアル テストのステップ内で他のビジュアル テストを再生している場合に役立ちます。

 **ヒント:** ローカル変数 ウィンドウからクリップボードに変数をコピーするには、変数を右クリックします。**コピー** を選択すると、現在選択している変数（ノード）とそのすべての子変数がコピーされます。また、**値のコピー** を選択すると、現在選択している変数の **値** だけがコピーされます。変数の **値** が空の場合は、このメニュー項目は無効になります。

ローカル変数

ローカル変数は、1 つのビジュアル テスト内で使用するために作成された変数です。これらはビジュアル テストの <<開始>> ステップで定義し、**プロパティ** ウィンドウで編集し、再生中にビジュアル テストで使用できます。ローカル変数を作成すると、変数を作成したビジュアル テストで使用できます。使用できる場所とタイミングは、以下のとおりです。


- テストロジック
- 結果コメント
- 式
- 他のビジュアル テストで設定されたグローバル変数を格納するとき

ビジュアル テストへのローカル変数の追加

ローカル変数は、1 つのビジュアル テスト内で使用するために作成された変数です。これらはビジュアル テストの <<開始>> ステップで定義し、**プロパティ** ウィンドウで編集し、再生中にビジュアル テストで使用できます。ローカル変数を作成すると、変数を作成したビジュアル テストで使用できます。

1. ローカル変数を追加するビジュアル テストを開きます。
2. **挿入 > 変数 > ローカルの追加** を選択します。**ローカル変数の追加** ダイアログ ボックスが開きます。
3. 変数の名前を **変数名** フィールドに入力します。

ローカル変数名に有効な文字は、大文字/小文字の英数字とアンダースコア（「ABC」、「abc」、「_」）です。また、ローカル変数名は一意にする必要があります。

 **注:** 変数名は、「st_」で始めることはできません。

追加または編集される変数の名前が重複する場合、Silk Test Workbench によってそれが通知されます。この場合、追加または編集する変数に固有の名前を付ける必要があります。


4. **型** リストから変数の型を選択します。

ローカル変数を追加または編集する場合、表示されるデータ型は、<<開始>> ステップのプロパティ ウィンドウを使用して追加しようとしている変数のデータに基づきます。式の結果を格納するとき、リストから型を選択します。

Silk Test Workbench は、次の変数の型をサポートします。

テキスト

変数の値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

 **注:** テキスト データ型は大文字/小文字を区別します。つまり、比較時にはテキスト文字列の個々の文字の大文字/小文字が一致していなければなりません。大文字/小文字が一致していない場合は、比較は失敗します。

数値 (Long)

変数の値は、-2,147,483,648 から 2,147,483,647 の範囲の数値です。（小数値または 10 進数値で使用される）ピリオドは使用できません。

数値 (Double)

変数の値は、倍精度浮動小数点数で、-1.7E308 から +1.7E-307 の 64 ビットの数値として格納されます。

ブール値 (True/False) 変数の値は True または False です。

数値 (Long Long) 64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。

列挙 この変数は一連の値をグループ化し、1 から n まで順番に番号を付けます。限られた個数の一意の値のみを変数に保持する場合は、列挙型を宣言します。この変数型は、列挙データ型を必要とするプロパティまたは変数に使用します。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

5. 省略可能: 変数の初期値を **初期値** フィールドに入力します。この値は、ビジュアルテスト、またはスクリプトに渡された場合はスクリプトで変更できます。値は型と整合性がある必要があります。たとえば、型が数値(Long Long)である場合、初期値には、整数でない文字 (テキスト文字など) を含めることはできません。ブール型変数の場合、リストから **False** または **True** を選択します。変数の場合、初期値は必須ではないため、このフィールドは空白のままにすることができます。この値は、ビジュアルテストの他の箇所で決定できます。
6. **OK** をクリックします。変数名と初期値 (存在する場合) が、<<開始>> ステップの **プロパティ** ペインの選択した変数型の下に表示されます。
7. 他のローカル変数を作成するには、必要に応じて上記のステップを繰り返します。各変数型の値領域は、現在定義されている各変数型の数を示します。たとえば、数値 (Long) 項目の値領域が **3 個の項目** の場合、現在このビジュアルテストには Long データ型のローカル変数が 3 つ定義されています。

ビジュアルテスト用ローカル変数の編集

ローカル変数は、<<開始>> ステップの **プロパティ** ペインで編集できます。

1. 編集するローカル変数が含まれているビジュアルテストを開きます。
2. ビジュアルテストの <<開始>> ステップを選択します。<<開始>> ステップのプロパティが **プロパティ** ペインに表示されます。
3. **プロパティ** ペインで、**変数** カテゴリを探るか **変数** をダブルクリックしてカテゴリを展開し、ビジュアルテストに定義されたローカル変数を表示します。

変数 カテゴリで定義される変数には、以下の型があります。

ビジュアルテスト 他のビジュアルテストで定義され、このビジュアルテストに渡される変数。

テキスト 変数の値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。



注: テキストデータ型は大文字/小文字を区別します。つまり、比較時にはテキスト文字列の個々の文字の大文字/小文字が一致していなければなりません。大文字/小文字が一致していない場合は、比較は失敗します。

数値 (Long) 変数の値は、-2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。

数値 (Double) 変数の値は、倍精度浮動小数点数で、-1.7E308 から +1.7E-307 の 64 ビットの数値として格納されます。


ブール値 (True/False) 変数の値は True または False です。

数値 (Long Long) 64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。

列挙 この変数は一連の値をグループ化し、1 から n まで順番に番号を付けます。限られた個数の一意の値のみを変数に保持する場合は、列挙型を宣言します。この変数型は、列挙データ型を必要とするプロパティまたは変数に使用します。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

4. 編集する変数の値領域をクリックし、値領域内の **変数の編集** ボタンをクリックします。変数の編集 ダイアログ ボックスが開きます。
5. 変数の名前を **変数名** フィールドに入力します。
ローカル変数名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。また、ローカル変数名は一意にする必要があります。

 **注:** 変数名は、「st_」で始めることはできません。

追加または編集される変数の名前が重複する場合、Silk Test Workbench によってそれが通知されます。この場合、追加または編集する変数に固有の名前を付ける必要があります。

6. 省略可能: 変数の初期値を **初期値** フィールドに入力します。この値は、ビジュアルテスト、またはスクリプトに渡された場合はスクリプトで変更できます。値は型と整合性がある必要があります。たとえば、型が数値(Long Long)である場合、初期値には、整数でない文字 (テキスト文字など) を含めることはできません。ブール型変数の場合、リストから **False** または **True** を選択します。変数の場合、初期値は必須ではないため、このフィールドは空白のままにすることができます。この値は、ビジュアルテストの他の箇所ですべて決定できます。
7. **OK** をクリックします。変数名と初期値 (存在する場合) が、<<開始>> ステップの **プロパティ** ペインの選択した変数型の下に表示されます。
8. 他のローカル変数を編集するには、必要に応じて上記のステップを繰り返します。
スクリプトに渡す変数を編集する場合は、変数値を渡されるテスト内でスクリプトを実行する必要があります。

ビジュアルテストからのローカル変数の削除

ローカル変数がビジュアルテストで必要なくなった場合は、ローカル変数を削除します。

1. ローカル変数を削除するビジュアルテストを開きます。
2. 次のいずれか 1 つのステップを行います:
 - すべてのローカル変数をテストから削除するには、<<開始>> ステップを選択して Del を押します。
 - 特定のローカル変数をテストから削除するには、<<開始>> ステップを選択し、**プロパティ** ペインで変数を選択して **削除** をクリックします。

グローバル変数

Silk Test Workbench では、ビジュアルテスト内のデータを表す変数を作成して、同じデータベース内の他のビジュアルテストからそのデータを使用できます。この場合は、グローバル変数にデータを設定します。

グローバル変数は、複数のテストで使用する必要のあるデータ、または定数として使用するデータに対して有効です。データをグローバル変数に定義すると、そのデータを必要とする個々のビジュアルテストで、そのグローバル変数を参照できます。グローバル変数で表されるデータを変更する場合は、データを 1 箇所ですべて変更します。各ビジュアルテストを次回再生したときに、自動的に新しいデータが使用されます。

グローバル変数を使用するには、1 つのビジュアルテストで変数を作成し、他のビジュアルテストでその値をローカル変数に格納します。これにより、1 つのテスト内にデータを作成し、テストソリューション

全体でそのテストを再利用できます。グローバル変数のデータがローカル変数に読み込まれると、そのデータはローカル変数と同じように使用できます。

ローカル変数とは異なり、グローバル変数は、その変数が作成されたビジュアル テストを再生するときだけでなく作成され、削除されるわけではありません。グローバル変数は、この変数が含まれるビジュアル テストを再生する任意のビジュアル テストから、アクセスおよび変更できます。

ビジュアル テストにグローバル変数を作成するには、グローバルの設定ステップを作成し、そのステップに変数を定義するプロパティを設定します。各グローバルの設定ステップは、1 つのグローバル変数定義を表します。

ビジュアル テストにおけるグローバル変数の管理

ビジュアル テストでグローバル変数値を使用するには、スクリプトに定義されたローカル変数にグローバル変数を格納し、ローカル変数として使用する必要があります。また、この変数が含まれるビジュアル テストを、値を使用するビジュアル テストに挿入する必要があります。これには、グローバル変数が含まれるビジュアル テストのすべてのステップが含まれますが、これにより不要なテスト結果が生成される場合があります。

このため、テストソリューションのすべてのグローバル変数は、1 つのビジュアル テストに作成して管理することをお勧めします。これには以下のような利点があります。

- テストソリューションで使用する各グローバル変数定義を 1 箇所を確認できます。
- グローバル変数名の重複を回避できます。
- グローバル変数が含まれるビジュアル テストを他のビジュアル テストに挿入した場合、再生時にグローバル変数のみが呼び出し元のスクリプトに読み込まれます。グローバル変数が含まれるビジュアル テストで、他の不要なステップが再生されません。

ローカル変数を使用してグローバル変数を格納すると、格納前にそのローカル変数に設定されていた値は、グローバル変数の値に変更されます。

ビジュアル テストに定義されたグローバル変数と同じ名前のローカル変数を定義できます。グローバル変数の値を変更した場合でも、同じ変数名を持つローカル変数の値は変更されません。ただし、ビジュアル テストで誤ってデータを使用したり混乱したりすることのないように、グローバル変数名はローカル変数名や他の変数名とは異なる名前にすることをお勧めします。

2 つのビジュアル テストで同じ名前のグローバル変数を作成することもできますが、グローバル変数に予想外の値が格納されたまま使用されるおそれがあるため、このような使用法も避けます。

ビジュアル テスト用グローバル変数の作成

グローバル変数は 1 つのビジュアル テスト内で使用するために作成しますが、他のビジュアル テストからも使用できます。グローバル変数は、グローバルの設定ステップに **割り当て** プロパティを使用して定義します。グローバル変数を作成すると、このグローバル変数が定義されているビジュアル テストを実行するビジュアル テストで、変数を使用できます。

1. グローバル変数を作成するビジュアル テストを開きます。
2. グローバル変数を作成する場所の直前のステップを選択します。
3. **挿入 > 変数 > グローバルの設定** を選択します。選択したステップのあとにグローバル変数の設定ステップが作成されます。グローバル変数の設定ステップのプロパティが、**プロパティ** ペインに表示されます。
4. グローバル変数の設定ステップの **プロパティ** ペインで、プロパティをカテゴリごとにグループ化します。
5. **割り当て** カテゴリ プロパティを更新して、グローバル変数の名前およびグローバル変数に割り当てる値を定義します。**割り当て** カテゴリ プロパティを更新し、ビジュアル テストを保存すると、グローバル変数の設定ステップは「グローバル変数 'グローバル変数名' に '値' を設定します」となります。
6. 他のグローバル変数を作成するには、必要に応じてステップ 2 から 5 を繰り返します。グローバル変数を設定すると、他のビジュアル テストのローカル変数に読み込むことができます。このためには、まず

グローバル変数が定義されたビジュアルテストを、ローカル変数が定義されたビジュアルテストに挿入する必要があります。

ビジュアルテストで使用するグローバル変数の取得

グローバルの設定ステップでグローバル変数が作成されると、それらのグローバル変数は、グローバルの設定ステップを含むビジュアルテストを実行する任意のビジュアルテストで使用できます。グローバル変数は、まずこのグローバル変数を使用するビジュアルテストに作成されたローカル変数に読み込んでから、ローカル変数として使用する必要があります。

1. グローバル変数を使用するビジュアルテストを開きます。
2. グローバル変数値を格納するローカル変数を作成します。
ローカル変数は任意のテストステップに作成できます。
3. グローバル変数を作成したビジュアルテストを実行するステップを作成します。同じビジュアルテストに定義されているローカル変数にグローバル変数を取得する場合は、このステップをスキップしてください。
これにより、グローバル変数を使用するビジュアルテストから、グローバル変数にアクセスできます。
4. グローバル変数を取得する場所の直前のステップを選択します。
5. **挿入 > グローバル > グローバルの取得** を選択します。選択したステップのあとにグローバル変数の内容の取得ステップが作成されます。グローバル変数の内容の取得ステップのプロパティが **プロパティ** ペインに表示されます。
6. グローバルの取得ステップの **プロパティ** ペインで、プロパティをカテゴリごとにグループ化します。
7. **割り当て** カテゴリプロパティを更新して、グローバル変数の名前およびグローバル変数に割り当てる値を定義します。プロパティを更新してビジュアルテストを保存すると、グローバル変数の内容の取得ステップが、「グローバル変数 '[グローバル変数名]' の内容を取得して、ローカル変数 '[ローカル変数名]' に格納します」のように表示されます。
8. 他のグローバル変数を作成するには、ステップ 2 から 7 を繰り返します。グローバル変数をローカル変数に取得すると、ローカル変数にグローバル変数の値が格納されます。この値をビジュアルテストの式で使用したり、同様にローカル変数として使用したりします。

予約変数

Silk Test Workbench では、ビジュアルテストの再生に関するデータを公開し、このデータをビジュアルテストの再生時に使用できます。このデータには、再生結果全体や特定の再生結果の合格/失敗の情報、コントロールに入力された現在値、最後に発生したエラーのテキスト、最後に実行されたステップの情報などが含まれます。公開された再生データには、さまざまな用途があります。たとえば、実行されたステップの成否に基づいて条件を作成できます。この条件を再生時の結果への対処を決定するために使用できます。

ビジュアルテストの再生データは、予約変数に格納されます。予約変数は、Silk Test Workbench によって定義されている変数です。この変数は読み取り専用です。つまり、その値をビジュアルテストで返して、使用することができますが、ユーザーが値を設定することはできません。予約変数は、必ず接頭部「st_」で始まります。

予約変数は、値として変数を使用可能な任意のプロパティの値として返すことができます。

予約変数は、同じテスト内で実行されるビジュアルテスト間で双方向に継承されます。たとえば、ビジュアルテスト B をビジュアルテスト A に挿入すると、ビジュアルテスト B はビジュアルテスト A の予約変数値を継承します。ビジュアルテスト B を再生したあと、ビジュアルテスト A はビジュアルテスト B の予約変数値を継承します。

ビジュアルテスト内で使用する予約変数を選択するには、**変数の選択** ダイアログボックスを使用します。

予約変数の一覧

以下の表に、ビジュアルテストの再生に関連するデータを返す予約変数を示します。

変数名	デフォルト値	タイプ	説明
st_LastError	空の文字列	テキスト	再生時に最後に生成されたエラーのテキストを返します。再生時 (st_LastError 値が返される場所まで) にエラーが生成されなかった場合、この値は Nothing です。
st_LastReturnValue	False	各種	ビジュアル テストで最後に実行されたステップの値を返します。戻り値は、ステップ タイプによって異なります。 ほとんどの自動ステップは、操作が成功すると True を返し、失敗すると False を返します。文字列、列番号、日付などの値を返す自動ステップまたはその他のステップでは、st_LastReturnValue 変数にその情報を返します。
st_LastValidation	False	ブール値	最後に実行された検証ロジックの結果を返します。 True = 最後に実行された検証ロジックは成功しました。 False = 最後に実行された検証ロジックは失敗しました。
st_Project	実行中のビジュアル テストが格納されているプロジェクトの名前	テキスト	実行中のビジュアル テストが格納されているプロジェクトの名前を返します。
st_RepeatListValue	空の文字列	テキスト	最後にアタッチされたコントロールの列挙が実行されたとき、そのコントロールの現在の入力値を返します。たとえば、繰り返しロジックを使用して ListBox コントロールの項目を繰り返し検索する場合、この変数は ListBox 内の現在の項目を返します。 この値は、コントロールに含まれる項目の名前を返します。繰り返しロジックを使用してリスト内の項目を繰り返し検索し、選択する値として st_RepeatListValue を使用します。
st_Username	現在のユーザー名	テキスト	ビジュアル テストまたは .NET スクリプトを実行しているユーザーの名前を返します。
st_VisualTest	実行中のビジュアル テストの名前	テキスト	実行中のビジュアル テストの名前を返します。

予約変数の戻り値

予約変数およびさまざまなステップ タイプに対して返される値を以下に示します。

ステップ タイプ	st_LastReturnValue	st_LastError	st_LastValidation	st_RepeatListValue
<<開始>>	True	変更なし	変更なし	変更なし
<<終了>>	True	該当なし	該当なし	該当なし

ステップタイプ	st_LastReturnValue	st_LastError	st_LastValidation	st_RepeatListValue
自動ステップ	Open Agent メソッドの戻り値。	メソッドが設定したエラーメッセージが格納されます。	変更なし	変更なし
ビジュアルテストの再生	True = ビジュアルテストが実行されました。 False = ビジュアルテストが実行されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
検証	最後に実行された検証ロジックの結果を返します。 True = 最後の検証に成功しました。 False = 最後の検証に失敗しました。	すでに存在しない変数の参照など、評価された式が失敗した場合にエラーメッセージが格納される場合があります。	変更なし	変更なし
グローバル変数の内容の取得	True = 変数の内容を取得しました。 False = 内容が取得されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
グローバル変数の設定	True = 変数の内容が設定されました。 False = 変数の内容が設定されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
再生設定の取得	True = 再生設定がローカル変数に入力されました。 False = 再生設定がローカル変数に入力されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
再生設定の設定	True = 再生設定が設定されました。 False = 再生設定が設定されませんでした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし

ステップ タイプ	st_LastReturnValue	st_LastError	st_LastValidation	st_RepeatListValue
プログラムの実行	ShellExecute API 関数の戻り値 (32 より大きければ成功)。これは VBA Shell() 関数によく似ています。	st_LastReturnValue > 32 : 空です。 st_LastReturnValue <= 32 : Windows から返されるエラー メッセージが格納されます。たとえば、2 の場合、メッセージは「指定されたファイルが見つかりません。」となります。	変更なし	変更なし
メッセージ ボックス	押されたボタンを表す数値 (Long)。 1 = OK 2 = キャンセル 3 = 中止 4 = 再試行 5 = 無視 6 = はい 7 = いいえ	すでに存在しない変数の参照など、評価された式が失敗した場合にエラー メッセージが格納される場合があります。	変更なし	変更なし
開始タイマー	True = タイマーは開始されます。 False = タイマーはすでに開始されています。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラー メッセージが格納されます。	変更なし	変更なし
再開タイマー	True = タイマーは停止タイマーのあと、再開されます。 False = タイマーはすでに実行中です。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラー メッセージが格納されます。	変更なし	変更なし
タイマーのリセット	True	空	変更なし	変更なし
停止タイマー	指定されたタイマーの経過時間 (ミリ秒単位、Long)。タイマーが起動していない場合は False (ブール)。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラー メッセージが格納されます。	変更なし	変更なし
遅延	True	空	変更なし	変更なし
ラベル [ラベル名] へ移動	True = ステップが実行されました。	st_LastReturnValue = True : 空です。	変更なし	変更なし

ステップ タイプ	st_LastReturnV alue	st_LastError	st_LastValidation	st_RepeatListVal ue
エラー処理	False = ステップ が実行されません でした。 True = ステップ が実行されまし た。 False = ステップ が実行されませ ん でした。	st_LastReturnValue = False : エラー メッセージ が格納されます。 st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラー メッセージ が格納されます。	変更なし	変更なし
結果コメント	True = ステップ が実行されまし た。 False = ステップ が実行されませ ん でした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラー メッセージ が格納されます。	変更なし	変更なし
Using [ロケータ 名] (画面ステップ)	True = ステップ が実行されまし た。 False = ステップ が実行されませ ん でした。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラー メッセージ が格納されます。	変更なし	変更なし
項目プレースホルダ の破損	False	空	変更なし	変更なし
ロジック (If/ElseIf/ Else/EndIf)	変更なし	すでに存在しない変数の 参照など、評価された式が 失敗した場合にエラー メ ッセージが格納される場 合があります。	変更なし	変更なし
ループ (Repeat/ EndRepeat)	変更なし	すでに存在しない変数の 参照など、評価された式が 失敗した場合にエラー メ ッセージが格納される場 合があります。	変更なし	ループがリスト コ ントロールの内容を反 復しない場合は変更 なし。 ループがリスト型コ ントロールの内容を 反復する場合は、項 目のインデックスと 値の両方が複合値と して格納されます(ビ ジュアルテストでは 書式設定文字列とし て表されます)。
停止	True	空	変更なし	変更なし
ラベル	変更なし	変更なし	変更なし	変更なし

ステップタイプ	st_LastReturnValue	st_LastError	st_LastValidation	st_RepeatListValue
コントロールのプロパティ	True = コントロールが見つかって変数が正しく設定されました。 False = コントロールが見つからないか、ターゲット変数が見つからないなどのエラーが発生しました。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	常にプロパティの値に設定されます（ローカル変数が指定されない場合でも）。
eCATT 引数の内容の取得	引数の内容を取得した場合は True、内容を取得できなかった場合は False。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし
eCATT 引数の設定	引数の内容を設定した場合は True、そうでない場合は False。	st_LastReturnValue = True : 空です。 st_LastReturnValue = False : エラーメッセージが格納されます。	変更なし	変更なし

ビジュアルテスト間でのデータの共有

Silk Test Workbench では、ビジュアルテストから別のビジュアルテストにデータを渡すことができます。1つのテストでデータを格納するパラメータを作成します。次に、このテストを別のビジュアルテストで実行し、パラメータを渡してそのデータを使用します。

パラメータは、このパラメータが含まれるテストを実行する、任意のビジュアルテストに渡されます。パラメータが渡されると、データ値を割り当てたり、式、テストロジック、またはパラメータが使用可能な任意の場所でパラメータを使用したりできます。また、パラメータ値をローカル変数に割り当て、このローカル変数を使用することもできます。

パラメータを作成して、任意のビジュアルテストでデータを使用できるようにします。パラメータを使用するビジュアルテストで、以下の作業を実行します。

- パラメータを作成するビジュアルテストを挿入します。
- パラメータに値を割り当てます。

ビジュアルテストへのパラメータの追加

ビジュアルテストは、入力パラメータを使用してスクリプトまたは他のビジュアルテストからデータを受け取ったり、出力パラメータを使用して他のビジュアルテストにデータを渡したりできます。パラメータは、ビジュアルテストの <<開始>> ステップで定義し、**プロパティ** ウィンドウで編集し、再生中にビジュアルテストで使用できます。パラメータを作成すると、パラメータを作成したビジュアルテストで使用できます。

1. パラメータを作成するビジュアルテストを開きます。
2. **タスク** ペインで <<開始>> ステップをクリックします。
3. **プロパティ** ペインで、**変数** カテゴリに移動します。
4. 作成するパラメータの種類を選択します。

- 新しい入力パラメータを作成する場合は、**入力パラメータ** フィールドをクリックします。
- 新しい出力パラメータを作成する場合は、**出力パラメータ** フィールドをクリックします。

5. **パラメータの追加** をクリックします。**パラメータの追加** ダイアログ ボックスが開きます。

6. パラメータの名前を **名前** フィールドに入力します。

パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。

7. **型** リストからパラメータのデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。


ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ペリオドは使用できません。
数値 (Long Long)	64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。
列挙	このパラメータは一連の値をグループ化し、1 から n まで順番に番号を付けます。このパラメータ型は、列挙データ型を必要とするプロパティまたはパラメータに使用します。 列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

8. **初期値** フィールドに、渡されるパラメータの値を入力します。

空のままにした場合、ビジュアル テストで設定されたパラメータの初期値が使用されます。値は型と整合性がある必要があります。たとえば、型が数値 (Long) の場合、初期値は整数にする必要があります。


9. **OK** をクリックします。パラメータの名前と初期値 (存在する場合) が、<<開始>> ステップの**プロパティ** ペインの選択したパラメータ型の下に表示されます。


10 他のパラメータを作成するには、必要に応じて上記のステップを繰り返します。パラメータ型のフィールドは、その型の現在定義されているパラメータの数を示します。たとえば、数値 (Long) 項目のフィールドが **3 個の項目** の場合、現在このビジュアル テストには Long データ型のパラメータが 3 つ定義されています。

 **注:** パラメータが作成されたときには、パラメータに値は割り当てられていません。パラメータに値が割り当てられるのは、別のビジュアル テストでテストが実行されたときです。パラメータを作成したビジュアル テストを再生するときに、その値を割り当てる必要があります。このシナリオでは、再生を開始したときに **パラメータ** ダイアログ ボックスが開き、パラメータに値を割り当てることができます。

ビジュアル テストでのパラメータへの値の割り当て

ビジュアル テスト間でデータの受け渡しを行うには、あるビジュアル テストが他のビジュアル テストに内からテスト ステップとして呼び出される必要があります。以降、呼び出すビジュアル テストを **ホスト** テスト、呼び出されるビジュアル テストを **子テスト** と呼びます。実行時にホスト テストから子テストにパラメータを渡すには、ホスト テストを開き、子テストのパラメータに値を割り当てます。

 **注:** ビジュアルテストでパラメータに値を割り当てる前に、パラメータをビジュアルテストに追加する必要があります。詳細については、「[ビジュアルテストへのパラメータの追加](#)」を参照してください。

 **ヒント:** ビジュアルテスト間で渡されるパラメータの値を確認するには、結果コメントステップをビジュアルテストに追加します。コメントの**式**プロパティに対して**変数**を選択し、リストから、渡された値が格納されるパラメータを選択します。

1. ホストテストを開きます。
2. **テストステップ** ペインで、パラメータを含む子テストを再生するステップを選択します。
ステップテキストは、「ビジュアル テスト [資産名] を再生します」のようになります (ここで、[資産名] は子テストの名前です)。
ステップのプロパティが **プロパティ** ペインに表示されます。このプロパティには、ビジュアルテストパラメータとして設定されているパラメータが含まれます。パラメータは、**入力パラメータ** と **出力パラメータ** というカテゴリに分類されます。各パラメータは、データ型ごとにリストされます。各パラメータのプロパティ名がそのパラメータ名です。
パラメータが **プロパティ** ペインに表示されない場合は、子テストを再読み込みします。これを行うには、**名前** プロパティを選択し、**パラメータの更新** をクリックしてパラメータのリストを更新します。
3. パラメータを選択します。
4. **パラメータの編集** をクリックします。**パラメータの編集** ダイアログ ボックスが開きます。
5. パラメータの値を **初期値** フィールドに入力します。
6. 他のパラメータに値を割り当てるには、上記のステップを繰り返します。

ビジュアルテストのパラメータの編集

1. パラメータを作成するビジュアルテストを開きます。
2. **タスク** ペインで <<**開始**>> ステップをクリックします。
3. **プロパティ** ペインで、**変数** カテゴリに移動します。
4. 編集するパラメータの種類をに移動します。
 - 入力パラメータを編集する場合は、**入力パラメータ** に移動します。
 - 出力パラメータを編集する場合は、**出力パラメータ** に移動します。
5. 編集するパラメータを選択します。
6. **パラメータの編集** をクリックします。**パラメータの編集** ダイアログ ボックスが開きます。
7. パラメータの名前を **名前** フィールドに入力します。
パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。
8. **型** リストからパラメータのデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ペリオドは使用できません。
数値 (Long Long)	64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。

テキスト パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

列挙 このパラメータは一連の値をグループ化し、1 から n まで順番に番号を付けます。このパラメータ型は、列挙データ型を必要とするプロパティまたはパラメータに使用します。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

9. 初期値 フィールドに、渡されるパラメータの値を入力します。

空のままにした場合、ビジュアルテストで設定されたパラメータの初期値が使用されます。値は型と整合性がある必要があります。たとえば、型が数値 (Long) の場合、初期値は整数にする必要があります。

100K をクリックします。

ビジュアルテストからのパラメータの削除

1. 削除するパラメータが含まれているビジュアルテストを開きます。
2. **タスク** ペインで <<開始>> ステップをクリックします。
3. **プロパティ** ペインで、**変数** カテゴリに移動します。
4. 削除するパラメータのテキストボックスをクリックし、**削除** をクリックします。

Silk Central からのパラメータを使用する

To enable Silk Central でテストに対して設定されたパラメータを Silk Test Workbench で使用することができるようにするには、Silk Test Workbench で Silk Central のパラメータと同じ名前を持つ変数を定義し、そしてその変数を使用します。

ビジュアルテスト間でのデータの受け渡し

Silk Test Workbench では、ビジュアルテスト間でデータを受け渡すことができます。これにより、既存の Silk Test Workbench ビジュアルテストで作成されたデータを再利用できるだけでなく、テストソリューション内でデータの整合性を維持したまま、最も効果的なテスト資産を使用できるようになります。ビジュアルテスト間でのデータの受け渡しには、ビジュアルテストのパラメータを使用します。


ビジュアルテスト間でデータの受け渡しを行うには、あるビジュアルテストが他のビジュアルテストに内からテストステップとして呼び出される必要があります。以降、呼び出すビジュアルテストを **ホストテスト**、呼び出されるビジュアルテストを **子テスト** と呼びます。ビジュアルテスト間でデータを受け渡しを行うには、ビジュアルテストが同じプロジェクトに含まれているか、子テストが参照プロジェクトに含まれている必要があります。

ビジュアルテストのパラメータを使用すると、以下の作業を実行できます。


- ビジュアルテストでデータを作成し、それを他のビジュアルテストで使用する。
- ビジュアルテストでデータを作成し、それを子テストに渡して変更し、再びホストテストに戻す。

その手順は以下のとおりです。

1. 入力パラメータと出力パラメータを子テストに追加します。
2. 子テストをテストステップとしてホストテストに挿入します。
3. ホストテスト内で入力パラメータの値を編集することによって、ホストテストから子テストにデータを渡します。

 **注:** ホスト テストから送信する値は子テスト内に保存されませんが、子テストでの各種処理内で使用できます。


4. ホスト テストで、子テストの出力パラメータから渡されたデータを格納するローカル変数を作成します。
5. ホスト テストでローカル変数を使用します。


 **注:** 入力パラメータと出力パラメータを作成し、ビジュアル テストでこれらのパラメータを使用する方法の詳細な例については、「*SilkTest Workbench* ビジュアル テスト チュートリアルによるこそ」の「ビジュアル テストからのスクリプトの再生」を参照してください。

ビジュアル テスト間で受け渡すパラメータの定義


他のビジュアル テストからテスト ステップとしてビジュアル テストを呼び出すときに、これらのビジュアル テスト間でパラメータを受け渡すことができます。以降、呼び出すビジュアル テストをホスト テスト、呼び出されるビジュアル テストを子テストと呼びます。

- ホスト テストから子テストにパラメータを渡すには、子テストの入力パラメータとしてパラメータを指定する必要があります。
 - 子テストからホスト テストにパラメータを渡すには、子テストの出力パラメータとしてパラメータを指定する必要があります。
1. ホスト テストから子テストにパラメータを渡すには、パラメータを入力パラメータとして子テストに追加します。
 - a) 子テストを開きます。
 - b) <<開始>> ステップを選択します。
 - c) **プロパティ** ペインで、**入力パラメータ** を選択します。
 - d) **パラメータの追加** をクリックします。**パラメータの追加** ダイアログ ボックスが開きます。
 - e) **パラメータ名** フィールドに、新しいパラメータの名前を入力します。
 - f) **型** リストから、ビジュアル テストに渡すデータ型を選択します。
 - g) **初期値** フィールドの値は、デフォルト値のままにします。
 - h) **OK** をクリックします。**プロパティ** ペインの入力パラメータのリストに、入力パラメータが表示されます。

 **注:** ビジュアル テストに対して定義された入力パラメータは、ビジュアル テスト ステップには表示されません。入力パラメータは、<<開始>> ステップの**プロパティ** ペインで表示できます。

 **注:** ホスト テストから送信する値は、子テスト内に保存されませんが、子テストでの各種処理内で値を使用できます。

2. 子テストからホスト テストにパラメータを渡すには、パラメータを出力パラメータとして子テストに追加します。
 - a) 子テストを開きます。
 - b) <<開始>> ステップを選択します。
 - c) **プロパティ** ペインで、**出力パラメータ** を選択します。
 - d) **パラメータの追加** をクリックします。**パラメータの追加** ダイアログ ボックスが開きます。
 - e) **パラメータ名** フィールドに、新しいパラメータの名前を入力します。
 - f) **型** リストから、ビジュアル テストから渡すデータ型を選択します。
 - g) **初期値** フィールドの値は、デフォルト値のままにします。
 - h) **OK** をクリックします。**プロパティ** ペインの出力パラメータのリストに、出力パラメータが表示されます。

 **注:** ビジュアル テストに実際に宣言されている場合を除き、ビジュアル テストで定義された出力パラメータはビジュアル テストには表示されません。出力パラメータは、<<開始>> ステップの**プロパティ** ペインで表示できます。

- i) ホスト テストを開きます。


- j) 子テストからホスト テストに渡すパラメータに対してローカル変数を追加します。
- 子テストの最新のパラメータを表示するには、子テストの **再生** ステップの **プロパティ** ペインにある **名前** テキスト ボックスをクリックし、**パラメータの更新** をクリックします。

ビジュアル テストでの他のビジュアル テストから受け取ったパラメータの使用

あるビジュアル テストで、そこから呼び出す他のビジュアル テストの出カパラメータの値を格納できません。以降、呼び出すビジュアル テストを **ホスト テスト**、呼び出されるビジュアル テストを **子テスト** と呼びます。ホスト テストで子テストのパラメータを使用するには、子テストの出カパラメータをホスト テストのローカル変数に割り当てます。

- ホスト テストを開きます。
- ホスト テストに渡す子テストの出カパラメータそれぞれに対してローカル変数を作成します。
詳細については、「[ビジュアル テストへのローカル変数の追加](#)」を参照してください。
- テスト ステップ** ペインで、子テストを実行するステップを選択します。ステップ テキストは、「ビジュアル テスト '資産名' を再生します」のようになります (ここで、「資産名」は子テストの名前です)。
- ステップの **プロパティ** ペインで、**出カパラメータ** カテゴリを展開します。
子テストの出カパラメータの名前は、**出カパラメータ** サブカテゴリに表示されます。
パラメータが **プロパティ** ペインに表示されない場合は、**名前** テキスト ボックスをクリックして **パラメータ名の更新** をクリックします。
- 出カパラメータ名のテキスト ボックスをクリックし、リストから適切なローカル変数を選択します。
- ホスト テストに渡す子テストの出カパラメータそれぞれに対して前の手順を繰り返します。
- 省略可能 : 子テストからホスト テストに渡される出カパラメータの値を確認する場合は、各パラメータに対して **結果コメント** ステップをホスト テストに挿入します。
 - ホスト テストで、**結果コメント** を挿入する前のステップを右クリックします。
たとえば、子テストを呼び出すステップをクリックします。
 - 挿入 > 結果コメント** を選択します。
 - 新しいステップのプロパティで、**式** テキスト フィールドをクリックします。
 - 選択** をクリックします。
 - リストから **変数** を選択します。**変数の選択** ダイアログ ボックスが開きます。
 - 子テストの出カパラメータに対応する変数を選択します。
- ローカル変数を使用するように、ホスト テストの適切なテスト ステップを変更します。
 - テスト ステップ** ペインでステップを選択します。
 - プロパティ** ペインで、**パラメータ** セクションを展開します。
 - ローカル変数のプロパティの種類に対応したテキスト ボックスをクリックします。
ローカル変数が **テキスト** 型の場合は、たとえば、AUT のテキスト フィールドにテキストを入力する **テスト ステップ (typeKeys)** を選択し、**keys** テキスト ボックス (**テキスト** 型) をクリックします。
 - 選択** をクリックして、**変数** を選択します。**変数の選択** ダイアログ ボックスが開きます。
 - 使用するローカル変数を選択し、**OK** をクリックします。

再生するホスト テストに子テストが含まれていて、その入カパラメータと出カパラメータが正しく定義されていれば、ホスト テストを再生すると、ホスト テストと子テストの間でパラメータが正しく受け渡しされます。

 **注:** 入カパラメータと出カパラメータを作成してビジュアル テストで使用方法の詳細な例については、「[SilkTest Workbench ビジュアル テスト チュートリアル](#)による」の「[ビジュアル テストからのスクリプトの再生](#)」を参照してください。

テスト フローの管理

ビジュアル テストの再生中、通常、ステップは Silk Test Workbench によって順番に実行されます。ただし、ステップを順番に実行しない方が便利な場合もあります。たとえば、ビジュアル テストを完了する前

に、ビジュアルテスト内の複数のポイントで一連のステップを実行する必要がある場合などです。テスト内の複数の場所で一連のステップを繰り返す代わりに、テストの最後の一連のステップを配置して、各ポイントからそこに移動することができます。

特定のステップへの移動は、一定の条件が満たされた場合に繰り返される繰り返しロジックを終了する手段としても使用できます。

ビジュアルテスト内でラベルを使用して特定の領域を「ブックマーク」すると、テストのナビゲーションに使用できます。ビジュアルテストで定義された任意のラベルに移動するテストステップを作成できます。ラベルステップはビジュアルテスト内での単純なマーカーであるため、**テストステップ** ペインに表示される後続の一連のステップを説明する手段としても使用できます。

ビジュアルテスト内でのスクリプトの実行

ビジュアルテストでは、既存の .NET スクリプトを呼び出して実行できます。これにより、アプリケーションのテストに対するモジュール式アプローチが可能になり、テストの実行をより制御できるようになります。ビジュアルテスト内でスクリプトを実行する場合は、テストサイトの整合性を保ち、テスト対象のテストアプリケーションが正しい初期再生状態になるようにすることが重要です。

この手順を使用してドライバテストを作成することもできます。

1. スクリプトを挿入するビジュアルテストを開きます。
2. 挿入したスクリプトを実行するポイントの前のテストステップを選択します。
3. **挿入 > .NET スクリプト** を選択します。 **.NET スクリプトの参照** ダイアログ ボックスが開きます。
4. 挿入するスクリプトを選択し、**OK** をクリックします。 Silk Test Workbench によって、選択したステップの下にステップが挿入されます。挿入されたステップによって、選択したスクリプトが呼び出され、再生されます。ステップテキストは、次のようになります：

```
.NET スクリプト '資産名' を再生します
```

ここで、'資産名' は、挿入されたスクリプトの名前です。

再生中、前のステップが実行されると、ビジュアルテスト内の次のステップが実行される前に、挿入されたスクリプトが最後まで再生されます。スクリプトがビジュアルテストに挿入されたあと、スクリプトを編集して、再生に使用される設定を構成できます。

テスト対象アプリケーションは、必要に応じて設定、フック、開始され、再生セッションの終了時までフックされたままになります。たとえば、ビジュアルテストにメモ帳用のアプリケーション構成があり、.NET スクリプトに電卓用のアプリケーション構成がある場合には、次のようになります。

- ビジュアルテストの実行の開始時に、メモ帳がフックされ、開始されます。
- 挿入された .NET スクリプトの実行の開始時に、電卓がフックされ、開始されます。メモ帳はフックされたままです。
- 挿入された .NET スクリプトの実行を停止しても、メモ帳と電卓の両方ともフックされたままになります。
- 再生セッションの終了時、つまり、ビジュアルテストの実行を停止すると、メモ帳と電卓の両方のフックが解除されます。

ビジュアルテストに挿入されたスクリプトの変更

スクリプトがビジュアルテストに挿入されたあと、スクリプトを編集して、再生に使用される設定を構成できます。

1. 変更するスクリプトを開き、変更し、保存します。
たとえば、スクリプトの入力パラメータまたは出力パラメータを変更する場合があります。
2. 変更したスクリプトを挿入したビジュアルテストを開きます。
3. **テストステップ** ペインで、スクリプトを実行するビジュアルテスト内のステップを選択します。ステップテキストは、「.NET スクリプト '資産名' を再生します」のようになります（ここで、資産名は挿入されたスクリプトの名前です）。

ステップのプロパティが **プロパティ** ペインに表示されます。ここでは、スクリプトの入力パラメータと出力パラメータが表示されます。

4. ステップの **プロパティ** ペインで、**.NET スクリプト** カテゴリを展開します。
挿入されたスクリプトの名前が **名前** 行に表示されます。
 - a) ビジュアル テストに挿入されているスクリプトを変更するには、**.NET スクリプトの参照** をクリックし、使用するスクリプトを選択します。
 - b) パラメータ名のみを更新するには、**パラメータ名の更新** をクリックし、プロンプトが表示されたら **はい** をクリックします。
 - c) 使用可能なパラメータを再読み込みするには、**パラメータの更新** をクリックし、プロンプトが表示されたら **はい** をクリックします。
 - d) 再生設定を変更するには、**再生設定** リストから別の値を選択します。

ビジュアル テスト内でのメッセージ ボックスの作成

ビジュアル テストでは、再生中にテスト担当者に対してメッセージを表示できます。たとえば、メッセージ ボックスを使用して、テスト出力を手動で検証したり、コントロールに関する情報を表示したり、テスト中にその他の任意の情報を通知したりできます。

メッセージ ボックスを作成するには、ビジュアル テストにメッセージ ボックス ステップを挿入します。再生中、メッセージ ボックスが表示され、ステップの作成時に指定したメッセージの内容が示されます。

1. メッセージを挿入するビジュアル テストを開きます。
2. メッセージ ボックスを表示するポイントの前のステップを選択します。
3. **挿入 > メッセージ ボックス** を選択します。Silk Test Workbench によって、選択したステップの下にメッセージ ボックスステップが挿入されます。
4. メッセージ ボックスステップの **プロパティ** ペインで、アイコンをクリックして、プロパティをカテゴリごとにグループ化するか、アルファベット順に並べ替えます。
5. **プロパティ** ペインで **全般** カテゴリのプロパティを編集し、メッセージ ボックスのタイトルとテキストを指定します。
6. **プロパティ** ペインで **詳細設定** カテゴリのプロパティを編集し、メッセージ ボックスの表示要素（ボタン、アイコンなど）を指定します。

ビジュアル テスト ナビゲーション用のラベル テスト ステップの作成

ラベル テスト ステップでは、テスト操作は実行されません。プログラム フロー管理用のラベル テスト ステップを使用して、「移動」ステップの移動先として使用すると共に、後続の一連のステップを記述します。

1. ラベルを挿入するビジュアル テストを開きます。
2. ラベル ステップが必要なポイントの前のステップを選択します。
3. **テスト ステップ** ペインで、**操作 > 挿入 > ラベル** をクリックします。選択したステップのあとにラベル ステップが作成されます。ステップの **プロパティ** ウィンドウには、ラベル ステップの **ラベル** プロパティが表示されます。
4. このラベル ステップの **ラベル** プロパティの値領域をクリックして、ラベルの説明テキストを入力します。

説明テキストを使用して、ラベル ステップの用途を示します。ラベルを「移動」ステップの移動先として使用する場合は、プログラム フローがこのステップに分岐する理由を示すテキストを入力します。ラベルを使用して後続の一連のステップを説明する場合は、後続の一連のステップの目的を示すテキストを入力します。

ラベル プロパティの値を変更できます。ただし、そのラベル ステップを参照するステップ（ロジック ステップ、「移動」ステップなど）もすべて手動で更新して、ラベル ステップの新しい値を参照するようになる必要があります。

5. **保存** をクリックして、ラベル ステップとその **ラベル** プロパティの値を保存します。更新されたステップおよびその説明テキストがビジュアル テストに表示されます。

- ラベルステップをナビゲーション用に使用する場合は、必要に応じて、そのラベルステップに移動する「移動」ステップを作成します。

ビジュアルテストナビゲーション用の移動テストステップの作成

移動テストステップでは、テスト操作は実行されません。プログラムフロー管理用の移動テストステップを使用して、テストの再生中に指定したラベルのステップに移動します。

- 移動ステップを挿入するビジュアルテストを開きます。
- 移動ステップを挿入するポイントの前のステップを選択します。
- 挿入 > **プログラムフロー** > **移動** を選択します。選択したステップのあとに移動ステップが作成されます。ステップの**プロパティ** ウィンドウには、ラベルステップの**ラベル** プロパティが表示されます。
- 移動ステップの**ラベル** プロパティの値領域をクリックして、リストから移動先のラベルを選択します。



ヒント: 移動ステップの移動先としてラベルステップを使用するには、最初にラベルステップを作成します。

- 保存** をクリックして、移動ステップとその**ラベル** プロパティの値を保存します。更新されたステップおよびその説明テキストがビジュアルテストに表示されます。

ビジュアルテスト内での外部アプリケーションの起動

ビジュアルテストでは、再生中に外部アプリケーションを起動できます。これにより、テストアプリケーションが予想どおり起動するかどうかをテストしたり、起動中のアプリケーションをテストするときのテストサイトの状態を保つことができます。

- 外部アプリケーションを起動するビジュアルテストを開きます。
- アプリケーションを起動するポイントの前のステップを選択します。
- 挿入 > **外部プログラム** を選択します。Silk Test Workbench によって、選択したステップの下にプログラムの実行ステップが挿入されます。
- 新しく作成されたプログラムの実行ステップを選択して、**プロパティ** ペインにそのプロパティを表示します。**コマンド** カテゴリのプロパティが**プロパティ** ペインに表示されます。
- パス名** テキストボックスで、以下のいずれかのステップを実行します。

- 起動する外部プログラムのパスを入力します。
- ファイルの選択** をクリックして [ファイルの選択] ダイアログボックスを開き、起動する外部プログラムに移動します。
- 選択** をクリックし、外部プログラムを起動するために使用するリテラル値、変数、式、またはアクティブデータを指定します。



注: 実行する操作が**検索** または**探索** の場合、このプロパティではパス名のみが必要です。

- 操作** リストボックスから、アプリケーションの起動時に実行する操作を選択します。

以下のいずれか 1 つを選択できます。

- 開く** (デフォルト) : 指定したパラメータを使用して、指定ファイルを開きます。ファイルには、アプリケーション、ドキュメントファイル、またはフォルダを指定できます。
- 編集** : エディタを起動して、編集するファイルを開きます。**パス名** プロパティで指定されたファイルが、ドキュメントエディタで開くことのできるドキュメントファイルではない場合、この操作は実行されません。
- 探索** : Windows エクスプローラを使用して、**パス名** プロパティで指定されたフォルダを開きます。
- 検索** : 指定されたディレクトリを起点として、Windows エクスプローラを検索モードで起動します。ディレクトリは、**パス名** プロパティまたは**ディレクトリ** プロパティを使用して指定できます。**パス名** プロパティが使用され、かつ、ファイル名が含まれている場合は、テストを再生しても、このステップは実行されません。

- **印刷** : **パス名** プロパティで指定されたドキュメント ファイルを印刷します。 **パス名** プロパティで指定されたファイルがドキュメント ファイルではない場合、テストを再生してもこのステップは実行されません。
7. **パラメータ** テキスト ボックスで、起動のためにアプリケーションに渡すパラメータを指定します。
パス名 プロパティで指定されたファイルがアプリケーション ファイルの場合は、このプロパティを使用して、アプリケーションに渡して起動時に使用するパラメータを含む NULL 終端文字列を指定します。 **パス名** プロパティで指定されたファイルがドキュメント ファイルまたはその他のタイプのファイルの場合は、値を空白のままにします。
 8. **ディレクトリ** テキスト ボックスで、**操作** プロパティで指定された操作に使用できるデフォルト ディレクトリを指定します。
 9. **ウィンドウ スタイル** リスト ボックスから、起動時のアプリケーションまたはプログラムの表示方法を選択します。
値は以下のとおりです。
 - **非表示** : プログラムまたはアプリケーションが起動したとき、ウィンドウは非表示です。フォーカスは、他のウィンドウに渡されます。
 - **最大化** : アプリケーション ウィンドウを最大化します。
 - **最小化** : 起動時にアプリケーション ウィンドウを最小化し、Z オーダーで次に最上位にくるウィンドウをアクティブにします。
 - **復元** : アプリケーション ウィンドウを起動して表示します。ウィンドウが最小化または最大化されていた場合は、そのウィンドウの元のサイズと位置に復元されます。この値は、最小化されたウィンドウを復元するときに使用します。
 - **表示** : アプリケーション ウィンドウを起動し、現在のサイズと位置に表示します。
 - **デフォルトの表示** : アプリケーションを起動したプログラムからの情報に基づいて、ウィンドウの表示状態を設定します。
 - **表示の最大化** : アプリケーションを起動し、ウィンドウを最大化して表示します。
 - **表示の最小化** : アプリケーションを起動し、ウィンドウを最小化して表示します。
 - **表示を最小化し、アクティブ化しない** : 最小化されたウィンドウでアプリケーションを表示します。現在のアクティブ ウィンドウは、アクティブなままです。
 - **現在の状態で表示し、アクティブ化しない** : 現在のウィンドウ状態でアプリケーションを表示します。アクティブ ウィンドウは、アクティブなままです。
 - **最新の位置に表示し、アクティブ化しない** : 最後に使用したときのウィンドウ サイズと位置でアプリケーションを表示します。アクティブ ウィンドウは、アクティブなままです。
 - **通常表示** : アプリケーションを起動して、ウィンドウを表示します。最後に使用したウィンドウの位置と最小化または最大化の状態が復元されます。この値は、はじめてウィンドウを表示するときに使用します。

ビジュアル テストへのオブジェクトの待機ステップの挿入

ビジュアル テストの任意の場所にステップを挿入して、ビジュアル テストの再生中にオブジェクトが存在するようになるまで待機したり、オブジェクトが消滅するまで待機したりできます。ビジュアル テストの再生とテスト アプリケーションとの間で同期を取るために、オブジェクトの待機ステップを組み込みます。

1. オブジェクトの待機ステップを組み込むビジュアル テストを開きます。
2. 以下のいずれかのステップを実行します。
 - オブジェクトの待機ステップを実行する場所の直前のステップを選択し、**挿入 > 同期とタイミング > オブジェクトの待機** を選択します。
 - **ロジック ツールボックス** を開き、**このビジュアル テストの再生中に出現/消滅するオブジェクトを待機** を **テスト ステップ** ペインの適切な位置にドラッグします。

Silk Test Workbench によって、選択したステップの下に " が存在するまで待機します ステップが挿入されます。

3. 新しく作成された " が存在するまで待機します ステップを選択して、**プロパティ** ペインの **待機** カテゴリにそのプロパティを表示します。

4. **プロパティ** ペインの **ロケーター** テキスト ボックスで、待機するオブジェクトを識別します。

- **テスト対象アプリケーション**：テスト対象アプリケーションから直接可視のコントロールを識別するには、このボタンをクリックします。
- **画面プレビュー**：テスト対象アプリケーションが利用できない場合に、**画面プレビュー** から直接コントロールを識別するには、このボタンをクリックします。
- **オブジェクトの識別ダイアログ**：非表示コントロールを識別するために **オブジェクトの識別** ダイアログ ボックスを使用するには、このボタンをクリックします。**オブジェクトの識別** ダイアログ ボックスを使って、コントロールのロケーターを編集することもできます。
- **選択**：このボタンをクリックして、リテラル、変数、式の結果、またはアクティブ データ値を割り当てます。

5. **待機の種類** リストから、オブジェクトの存在を待機するか消滅を待機するかを指定します。

6. **タイムアウト** テキスト ボックスで、オブジェクトが存在または消滅するまで待機する最大時間をミリ秒で入力します。

デフォルトでは、この値は **オプション** ダイアログ ボックスの **デフォルトの待機タイムアウト (ミリ秒)** オプション値に設定されます。タイムアウト期間内に一致するオブジェクトがない場合は、再生エラーが発生します。

7. **保存** をクリックします。

プロパティの待機

テスト対象アプリケーションのコントロールのプロパティが、特定の値になるまで、または特定の値と異なる値になるまで待機するテスト ステップをビジュアル テストに追加できます。このようなステップは、ビジュアル テストの再生とテスト アプリケーションとの間で同期を取るために使用します。

1. ステップを追加するビジュアル テストを開きます。
2. ステップを挿入するポイントの前のステップを右クリックします。
3. Silk Test Workbench が待機する条件を選択します。
 - プロパティの値が特定の値になるまで待機する場合は、**挿入 > 同期とタイミング > プロパティの待機** を選択します。
 - プロパティの値が特定の値と異なる値になるまで待機する場合は、**挿入 > 同期とタイミング > プロパティの待機 (否定)** を選択します。

Silk Test Workbench によって、選択したステップの下に新しいステップが挿入されます。

4. 新しく作成されたステップを選択して、**プロパティ** ペインにそのプロパティを表示します。
5. **ロケーター** フィールドには、待機するプロパティを含む AUT のコントロールに対するロケーターを指定します。
6. **プロパティ** フィールドで、プロパティを選択します。
7. **値** フィールドに、待機する値を指定します。
8. **タイムアウト** フィールドには、再生の続行を待機する最大時間を指定します。
9. **保存** をクリックして、ステップとそのプロパティの値を保存します。更新されたステップおよびその説明テキストがビジュアル テストに表示されます。

コントロールの表示が安定するまで待機

テスト対象アプリケーション (AUT) のコントロールの表示が安定し、位置が変化しなくなるまで待機するテスト ステップをビジュアル テストに追加できます。このようなステップは、ビジュアル テストの再生とテスト アプリケーションとの間で同期を取るために使用します。

1. ステップを追加するビジュアル テストを開きます。
2. ステップを挿入するポイントの前のステップを右クリックします。
3. **挿入 > 同期とタイミング > スクリーンショットの安定を待機** を選択します。Silk Test Workbench によって、選択したステップの下に新しいステップが挿入されます。

4. 新しく作成されたステップを選択して、**プロパティ** ペインにそのプロパティを表示します。
5. **ロケーター** フィールドには、待機する AUT のコントロールに対するロケーターを指定します。
6. **タイムアウト** フィールドには、再生の続行を待機する最大時間を指定します。
7. **間隔** フィールドには、表示が安定するまでに要する時間を指定します。
8. **保存** をクリックして、ステップとそのプロパティの値を保存します。更新されたステップおよびその説明テキストがビジュアルテストに表示されます。

ビジュアルテストの再生遅延の設定

ビジュアルテストの任意の場所に、ビジュアルテストの再生を遅延させるための遅延ステップを挿入できます。遅延の時間を制御できます。ビジュアルテストに挿入できる遅延の数に制限はありません。ビジュアルテストの再生とテストアプリケーションとの間で同期を取るために遅延を使用します。

1. 再生遅延を設定するビジュアルテストを開きます。
2. 遅延ステップを実行するポイントの前のステップを選択します。
3. **挿入 > 同期とタイミング > 遅延** を選択します。Silk Test Workbench によって、選択したステップの下に 1 秒の遅延ステップが挿入されます。
4. 遅延の増分と時間を変更するには、新しく作成された 1 秒の遅延ステップを選択し、そのプロパティを **プロパティ** ペインに表示します。
 - a) **遅延タイプ** リストから、遅延間隔に対して **秒** または **ミリ秒** を選択します。
 - b) **遅延時間** テキストボックスで、遅延の時間を指定します。

リテラル値を入力するか、あるいは **選択** をクリックして変数、式の結果、またはアクティブデータ値を割り当てます。
5. **保存** をクリックします。

ビジュアルテストへのタイマーの挿入

ビジュアルテストの任意の場所にタイミングステップを挿入して、再生中にタイマーを開始、停止、または再開できます。タイマーを組み込むと、特定のステップを再生するのにかかる時間を特定できます。

1. 開始タイマー、停止タイマー、または再開タイマーを組み込むビジュアルテストを開きます。
2. タイマーを挿入する場所の直前のステップを選択し、**挿入 > 同期とタイミング** を選択して、以下のコマンドのいずれかを選択します。
 - **開始タイマー**：時間再生する一連のステップの最初のステップを指定するには、このオプションを選択します。
 - **停止タイマー**：時間再生する一連のステップの最後のステップを指定するには、このオプションを選択します。
 - **再開タイマー**：一連のステップのタイミングの設定を再開するには、このオプションを選択します。たとえば、いくつかのテストステップの時間を記録することが必要でない場合は、停止タイマーを挿入し、次に、タイミングに含めるステップに対して再開タイマーを挿入できます。

Silk Test Workbench によって、選択したステップの下にタイマー番号 n の開始、タイマー番号 n の停止、またはタイマー番号 n の再開ステップが挿入されます。

3. 新しく作成されたステップを選択し、そのプロパティを **プロパティ** ペインの **タイマー** カテゴリに表示します。
4. **タイマーの動作** リストから、使用するタイマーの種類を選択します。
5. **タイマー番号** テキストボックスで、タイマーに割り当てる番号を入力するか、リストから選択します。複数の条件の時間を計るときに、各タイマーに一意的な番号を割り当ててタイマーを区別できます。ビジュアルテストには、最大で 10 個のタイマーを組み込むことができます。
6. **保存** をクリックして、ステップとそのプロパティの値を保存します。更新されたステップおよびその説明テキストがビジュアルテストに表示されます。

ビジュアルテストを再生すると、**結果** ウィンドウの **詳細** タブで、テストステップの **結果の詳細** 列にタイムマーの結果が表示されます。

テスト ステップ

ステップ（またはテスト ステップ）は、ビジュアル テストにおける主な実行可能単位です。ステップは、スクリプトのコード行に類似しています。

テスト ステップは、**ビジュアル ナビゲータ** の **テスト ステップ** ペインでのみ実行できます。

ビジュアル テストのステップ フローとロジックは、スクリプトのコード行のフローとロジックに似ています。たとえば、ビジュアル テストでは、コントロールに対して操作を実行する前に、テスト アプリケーションのそのコントロールにアタッチする必要があります。スクリプトと同様、ビジュアル テストにも、テスト アプリケーションのコントロールを表すオートメーション オブジェクトが含まれています。このオートメーション オブジェクトによって、コントロールに対して実行可能なプロパティとメソッドが公開されます。オートメーション操作の 1 つ 1 つが、ビジュアル テストの 1 ステップとして表現されます。

複数のステップの選択

ビジュアル テストのステップをグループとして選択すると、そのステップのグループごとに情報を更新できます。

1. 連続したテスト ステップのグループを選択するには、以下のステップを実行します。
 - a) 連続したステップのグループで最初のテスト ステップを選択します。
 - b) Shift キーを押したまま、連続したグループの最後のステップを選択します。または、**編集 > すべて 選択** を選択して、ビジュアル テストのすべてのステップを選択します。
2. 連続していないテスト ステップのグループを選択するには、以下のステップを実行します。
 - a) ステップのグループで最初のテスト ステップを選択します。
 - b) Ctrl キーを押したまま、任意にステップを選択します。ステップが連続している必要はありません。

テスト ステップのコピーと貼り付け

<<開始>> ステップと <<終了>> ステップ以外の任意のテスト ステップは、開いているビジュアル テストにコピーして貼り付けできます。テスト ステップをコピーおよび貼り付ける場合は、以下の点を考慮してください。

- Silk Test Workbench では、テスト ステップは、選択した順序ではなく、ビジュアル テストに表示されている順序でクリップボードにコピーされます。たとえば、テスト ステップ 5 を選択してからテスト ステップ 2 を選択し、その両方をクリップボードにコピーした場合は、ビジュアル テストにテスト ステップを貼り付けると、Silk Test Workbench によってテスト ステップ 2、テスト ステップ 5 の順に貼り付けられます。
- 画面テスト ステップをコピーしても、関連するテスト ステップはコピーされません。
- ラベルテスト ステップをコピーして貼り付ける場合は、ラベルテスト ステップが必ず一意になるように Silk Test Workbench によって「_1」が追加されます。
- 繰り返しテスト ステップまたは繰り返しの終了テスト ステップをコピーすると、関連する繰り返しの終了テスト ステップまたは繰り返しテスト ステップもコピーされます。
- If テスト ステップまたは End If テスト ステップをコピーすると、関連するすべての判断ロジック テスト ステップ (If、Else、Else If、End If) がコピーされます。ただし、判断ロジック テスト ステップに含まれる依存テスト ステップはコピーされません。
- 複数の Else テスト ステップまたは Else If テスト ステップは、すべての判断ロジック テスト ステップ (If、Else、Else If、End If) をコピーしないかぎり、クリップボードにコピーできません。Else If テスト ステップを貼り付ける場合、Silk Test Workbench で貼り付けできるのは、正しいコンテキスト内にあるステップのみです。

1. **テスト ステップ** ペインでビジュアル テストを開き、コピーするステップを選択します。

連続した、または連続していない複数のステップを選択して、そのステップを連続したグループとしてコピーし、貼り付けることができます。

2. 編集 > コピー を選択します。

コピーしたステップを別のビジュアル テストに貼り付けるには、そのビジュアル テストを開き、コピーしたステップを貼り付けます。コピーではなく、選択したステップを移動するには、**編集 > 切り取り** を選択します。

3. コピーしたステップを貼り付ける場所の直前のテスト ステップを選択します。

4. 編集 > 貼り付け を選択します。 Silk Test Workbench によって、ビジュアル テストの選択したステップのあとに、コピーしたテスト ステップが貼り付けられます。

画面プレビューを使用したテスト ステップの削除

画面プレビュー 内のキャプチャされたテスト アプリケーションのイメージは、テスト ステップに関連付けられているため、キャプチャされたイメージを **画面プレビュー** で削除すると、関連付けられているテスト ステップも **テスト ステップ** ペインから削除できます。次のテスト ステップが関連します。

- 関連付けられているすべての自動テスト ステップ
- その他のタイプのすべての単一ノード テスト ステップ
- すべての検証テスト ステップ

削除するキャプチャ イメージに関連付けられた判断および繰り返しロジックのテスト ステップは、**画面プレビュー** から削除することはできません。これにより、ロジックが保持され、ビジュアル テストが正しく再生されます。判断および繰り返しテスト ステップは、**テスト ステップ** ペインから手動で直接削除する必要があります。

1. テスト ステップ ペインで、キャプチャされたイメージのテスト ステップから、削除するステップを選択します。画面プレビュー に、キャプチャされたイメージが表示されます。

2. キャプチャされたイメージを 画面プレビュー で選択します。

3. Del を押します。サムネイルとそれに関連するステップの削除を確認するメッセージが表示されます。

4. 次のいずれか 1 つのステップを行います：

- **テスト ステップ** ペインで、キャプチャされたイメージとそれに関連するテスト ステップを削除するには、**はい** をクリックします。
- **画面プレビュー** のキャプチャされたイメージのみを削除し、関連するテスト ステップを保持するには、**いいえ** をクリックします。
- **キャンセル** をクリックして、サムネイルを削除することなくメッセージを閉じます。

[テスト ステップ] ウィンドウを使用したテスト ステップの削除

<<開始>> ステップと <<終了>> ステップ以外のテスト ステップは、**テスト ステップ** ペインから削除できます。ただし、テスト ステップによっては、正しく実行できるかどうか他のテスト ステップに依存するため、以下のルールが適用されます。

- ロジックの種類が If のテスト ステップを削除すると、削除された If テスト ステップに関連付けられているすべての Else If、Else、または End If テスト ステップも削除されます。
- ロジックの種類が End If のテスト ステップを削除すると、削除された End If テスト ステップに関連付けられているすべての If、Else、または Else If テスト ステップも削除されます。
- ロジックの種類が Else または Else If のテスト ステップを削除すると、Else または Else If テスト ステップのみが削除されます。If テスト ステップと End If テスト ステップは削除されません。
- ロジックの種類が 繰り返しのテスト ステップを削除すると、繰り返しの終了 テスト ステップも削除されます。
- 画面にアタッチするテスト ステップまたは自動ステップのテスト ステップを削除すると、判断ロジックと繰り返しロジックのテスト ステップを除き、関連するステップがすべて削除されます。アタッチステップに関連付けられた自動ステップは削除されます。ただし、関連付けられた操作ステップの前に、同じ画面にアタッチする別のアタッチ ステップが存在する場合は、削除されません。

1. テスト ステップ ペインでビジュアル テストを開き、削除するテスト ステップを右クリックして **削除** を選択します。

連続した、または連続していない複数のステップを選択して、そのステップを連続したグループとして削除できます。

2. テスト ステップのいずれかが画面タイプのステップの場合は、画面項目とそれに関連するステップの削除を確認するメッセージが表示されます。
 - 選択したステップとそれに関連するステップを削除するには、**はい** をクリックします。
 - 選択したステップのみを削除し、関連するテスト ステップを削除しない場合は、**いいえ** をクリックします。
 - テスト ステップを削除しないでメッセージを閉じるには、**キャンセル** をクリックします。

ストーリーボードを使用したテスト ステップの削除

サムネイルはテスト ステップのグループと関連付けられているため、**ストーリーボード** を使用してサムネイルを削除すると、関連するテスト ステップも **テスト ステップ** ペインから削除することができます。次のテスト ステップが関連します。

- 関連付けられているすべての自動テスト ステップ
- その他のタイプのすべての単一ノード テスト ステップ
- すべての検証テスト ステップとイベント ロジック テスト ステップ

削除するキャプチャ イメージに関連付けられた判断および繰り返しロジックのテスト ステップは、**画面プレビュー** から削除することはできません。これにより、ロジックが保持され、ビジュアル テストが正しく再生されます。判断および繰り返しテスト ステップは、**テスト ステップ** ペインから手動で直接削除する必要があります。

1. **ストーリーボード** で、削除するテスト ステップに対応するサムネイルをクリックします。
2. Del を押します。サムネイルとそれに関連するステップの削除を確認するメッセージが表示されます。
3. 次のいずれか 1 つのステップを行います：
 - **テスト ステップ** ペインで、サムネイルとそれに関連するテスト ステップを削除するには、**はい** をクリックします。
 - **ストーリーボード** のサムネイルのみを削除し、関連するテスト ステップを保持するには、**いいえ** をクリックします。
 - **キャンセル** をクリックして、サムネイルを削除することなくメッセージを閉じます。

ビジュアル テストの比較

2 つの異なるビジュアル テストや、同じビジュアル テストの 2 つのバージョンを比較して、これらの間の相違点を調べることができます。

2 つの異なるビジュアル テストや、同じビジュアル テストの 2 つのバージョン間の相違点のマージについては、「[ビジュアル テストのマージ](#)」を参照してください。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面** の **入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **ビジュアル テスト** を選択します。選択したプロジェクトの既存のビジュアル テストが、右側のペインに表示されます。



ヒント: 調査したいビジュアル テストが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 2 つの異なるビジュアル テストを比較するには：
 - a) **Ctrl + クリック** を使用して、比較する 2 つのビジュアル テストを選択します。
 - b) 選択した項目を右クリックします。
 - c) **相違点の表示** をクリックします。
4. 同じビジュアル テストの 2 つのバージョンを比較するには：
 - a) 調査したいビジュアル テストを右クリックして、**すべてのバージョンを表示** を選択します。


- b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する 2 つのバージョンを選択します。
- c) **相違点の表示** をクリックします。

- 変更箇所は黄色で表示されます。
- 追加箇所は緑色で表示されます。
- 比較した資産の一方に存在しない項目はグレーで表示されます。

ビジュアルテストのマージ


2 つの異なるビジュアルテストや、同じビジュアルテストの 2 つのバージョンを比較して、ビジュアルテスト間、またはビジュアルテストのバージョン間の相違点をマージすることができます。


1. **表示 > アセットブラウザ** を選択するか、**開始画面** の **入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストから **ビジュアルテスト** を選択します。選択したプロジェクトの既存のビジュアルテストが、右側のペインに表示されます。

 **ヒント:** 調査したいビジュアルテストが表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。


3. 2 つの異なるビジュアルテストを比較するには：
 - a) **Ctrl + クリック** を使用して、比較する 2 つのビジュアルテストを選択します。
 - b) 選択した項目を右クリックします。
 - c) **相違点の表示** をクリックします。
4. 同じビジュアルテストの 2 つのバージョンを比較するには：
 - a) 調査したいビジュアルテストを右クリックして、**すべてのバージョンを表示** を選択します。
 - b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する 2 つのバージョンを選択します。
 - c) **相違点の表示** をクリックします。
5. 相違点の表示を移動するには、ツールバーの **次の相違点** と **前の相違点** を使用します。
6. 選択した相違点をマージするには：
 - **左側にコピー** をクリックすると、右側のペインに表示されているバージョンまたは資産の選択した相違点を、左側に表示されているバージョンまたは資産にマージします。
 - **右側にコピー** をクリックすると、左側のペインに表示されているバージョンまたは資産の選択した相違点を、右側に表示されているバージョンまたは資産にマージします。

<<開始>> ステップを選択して **左側にコピー** または **右側にコピー** を押すと、**相違点の表示 - <<開始>>** ダイアログが開き、選択した方向に対してマージする相違点を選択できます。**OK** をクリックして、**相違点の表示** ダイアログの変更を適用します。

 **注:** アプリケーション構成をマージする場合、アプリケーション構成の種類が同じであれば、アプリケーション構成内の個々の要素ごとにマージできます。アプリケーション構成が、Web アプリケーションと標準アプリケーションの場合など、その種類が異なる場合は、アプリケーション構成全体のみをマージできます。

 **注:** **If** ステートメントや **Repeat** ステートメントなどのテストロジックステートメントをマージする場合、マージを適用した場合にターゲットのビジュアルテストが正しく動作することを Silk Test Workbench が確認します。ステートメントのマージによって、ビジュアルテストが動作しなくなる場合には、Silk Test Workbench はエラーメッセージを表示し、変更はマージされません。

- 変更箇所は黄色で表示されます。
- 追加箇所は緑色で表示されます。
- 比較した資産の一方に存在しない項目はグレーで表示されます。

 **注:** Silk Test Workbench ユーザーが資産のアクティブバージョンを比較またはマージしている間、その資産はロックされ、他のユーザーは編集できなくなります。資産が他のウィンドウで編集用に既に関われている場合に、その資産のアクティブバージョンを比較またはマージしようとすると、読み

取り専用モードでそのバージョンが開かれます。読み取り専用かどうかは、そのバージョンのタイトルバーに表示されます。

スクリプトの拡張

このセクションでは、VB .NET スクリプトを変更して、より高度な機能を実行する方法について説明します。

スクリプト内からのスクリプトの参照


よく使用される関数のスクリプトを作成し、別のスクリプトからこのスクリプトを再利用することができます。そのためには、**.NET スクリプト参照の追加** コマンドを使用します。

スクリプトの参照の追加

1. 別のスクリプトからの再利用または参照する単一のスクリプトを作成します。
2. 再利用されるスクリプトを呼び出す起動元スクリプトを新規作成します。
3. **プロパティ** タブを展開します。
4. 右クリックして、**.NET スクリプト参照の追加** を選択します。
5. 使用するスクリプトを選択して、**OK** をクリックします。

スクリプトの例

次のサンプルには、別のスクリプトから参照される、*Calculator* というスクリプトが含まれています。

 **注:** 次のサンプルには `Main()` メソッドが含まれていないため、独自に実行することはできません。

```
Public Class Calculator
    Public Shared Function Add(left As Integer, right As Integer)
        Return left + right
    End Function

    Public Shared Function Subtract(left As Integer, right As Integer)
        Return left - right
    End Function

    Public Shared Function Multiply(left As Integer, right As Integer)
        Return left * right
    End Function

    Public Shared Function Divide(left As Integer, right As Integer)
        Return left / right
    End Function
End Class
```

別のスクリプトを作成し、参照として *Calculator* を追加して (**プロパティ** タブを展開し、右クリックして、**.NET スクリプト参照の追加** を選択して、**Calculator** を選択)、次のコードを使用します。

```
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        MsgBox("2 + 3 = " + Calculator.Add(2, 3).ToString())
    End Sub
End Module
```

記録中のスクリプトへの検証の追加

以下の操作を行って、スクリプトの記録中に検証を追加します。

1. 記録を開始します。
2. 検証するオブジェクトの上にマウスカーソルを移動して、**Ctrl+Alt** を押します。
モバイル Web アプリケーションを記録する場合は、オブジェクトをクリックして **検証の追加** をクリックすることもできます。
このオプションを実行すると、記録が一時的に停止され、**テスト ロジック デザイナ** ウィザードが表示されます。
3. ウィザードが開いたあと、検証するオブジェクトを選択します。複合プロパティ (**Position** など) を使用してフィールドを選択した場合は、以下のコードに類似したコードがスクリプトに追加されます：
`Workbench.Verify(New TextPosition(1, 0), .TextField("TextField").Position)`。
4. ウィザードの指示に従って処理を行い、**完了** をクリックしてウィザードを閉じ、記録を続行します。

コーディングによるスクリプトへの検証の追加

スクリプトの結果が予想された結果と一致するかどうかを判断するために、検証を挿入します。

1. 検証を組み込むスクリプトを開きます。
2. 以下の検証メソッドのいずれかを挿入します。
 - 期待値と実際の値を比較するには、以下のように入力します。
`Workbench.Verify(expected As Object, actual As Object)`
たとえば、`Workbench.Verify("red", "red")` は検証に合格します。一方、`Workbench.Verify("red", "green")` は検証に失敗し、メッセージ「実際：[green]; 予想： [red]」が表示されます。
 - 期待値と実際の値を比較して、コメントを追加するには、以下のように入力します。
`Workbench.Verify(expected As Object, actual As Object, comment As String)`
たとえば、`Workbench.Verify("red", "green", "checking colors")` は検証に失敗し、メッセージ「checking colors - 実際：[green]; 予想： [red]」が表示されます。
 - 期待値と実際の値を比較して、コメントを追加し、さらに検証が失敗した場合にスクリーンショットを結果ファイルに追加するには、以下のように入力します。
`Workbench.Verify(expected As Object, actual As Object, comment As String, verifyFlags As VerifyFlags)`
たとえば、`Workbench.Verify("red", "green", "checking colors", verifyFlags.ScreenshotOnFailure)` は検証に失敗し、メッセージ「checking colors - 実際：[green]; 予想： [red]」が表示され、スクリーンショットが結果ファイルに追加されます。
 - 予想結果によって返された値を検証するには、以下のように入力します。
`Workbench.Verify(condition As Boolean)`
たとえば、`Workbench.Verify(True)` は検証に合格します。一方、`Workbench.Verify(False)` は検証に失敗します。
 - 予想結果によって返された値を検証して、コメントを追加するには、以下のように入力します。
`Workbench.Verify(condition As Boolean, comment As String)`
たとえば、`Workbench.Verify(True, "Test Passed")` は検証に合格します。一方、`Workbench.Verify(False, "Test Failed")` は検証に失敗します。
 - 予想結果によって返された値を検証して、コメントを追加し、さらに検証が失敗した場合にスクリーンショットを結果ファイルに追加するには、以下のように入力します。
`Workbench.Verify(condition As Boolean, comment As String, verifyFlags As VerifyFlags)`

たとえば、Workbench.Verify(True, "Test Passed", verifyFlags.ScreenShotOnFailure) は検証に合格し、スクリーンショットは追加されません。一方、Workbench.Verify(False, "Test Failed", ScreenShotOnFailure) は検証に失敗し、スクリーンショットが結果ファイルに追加されます。

- IEnumerable オブジェクト（リストや配列など）について実際の値と期待値を比較するには、以下のように入力します。


```
Workbench.Verify(expectedEnumerable, actualEnumerable)
```


例：

```
Dim selectedItemsList = listBox.SelectedItems ' we assume that a list with the
items "red" and "blue" is returned
Dim expectedItemsList = New List(Of String)()
expectedItemsList.Add("red")
expectedItemsList.Add("blue")
```

```
Workbench.Verify(selectedItemsList, expectedItemsList) ' verification passes
```

```
Dim expectedItemsArray = New String() { "red", "blue" }
Workbench.Verify(selectedItemsList, expectedItemsArray) ' verification passes
```

 **注:** 2つの IEnumerable オブジェクトは、要素の数が同じであり、それらの要素が等しく、かつ順序が同じである場合に、等しいとみなされます。

 **注:** 浮動小数点数を含む数学演算では、2つの数値が、ユーザーの視点では等しいにもかかわらず、内部表現のために完全には同一ではないとみなされる場合があります。このため、浮動小数点数（単精度、倍精度）は、差異が 0.00001 未満である場合は等しいとみなされます。この値は、状況によっては正しくない場合があります。その場合は、代わりに Verify(result As Boolean) を使用して 2つの値を比較します。

スクリプトでのテストデータの再利用

スクリプトで使用できる再利用の種類を示します。

スクリプトにおける変数

Silk Test Workbench ではスクリプト言語として VB.NET が使用されるため、スクリプトで VB.NET 変数を使用できます。変数は名前を付けられたエイリアスで、再生中に使用および変更が可能なデータを含んでいます。

記録時に、コントロールに対するデータ入力はリテラルデータとして記録されます。たとえば、WPF TextBox コントロールに対して入力された値がある場合、その結果のコードは以下のようになります。

```
.WPFFTextBox("@automationId='CarType').SetText("Ford")
```

これは機能しますが、スクリプトでは TextBox コントロールのデータとして *Ford* という値以外を使用できません。ただし、*Ford* をそのデータを表す変数で置き換えた場合、変数内でどのようなデータが使用されても、それが TextBox コントロールに入力されます。

```
.WPFFTextBox("@automationId='CarType').SetText(sCarmake)
```

変数を使用すると、スクリプトで使用するデータが定数である必要がなくなるため、テストの柔軟性が高まります。

型を指定しないで変数を宣言した場合、Silk Test Workbench によって、型が指定されなかったことを示すメッセージが生成されます。Silk Test Workbench では、その場合 Object データ型がデフォルトで使用されます。許容される任意の VB.NET データ型を変数データ型として使用できます。スクリプトで VB.NET 変数を使用するときには、VB.NET 変数の宣言に関するルールが適用されます。

例：

```
Dim dSalesPrice As Double
```

```
Dim iQuantity As Integer
```

```
Dim sCarmake As String
```

Silk Test Workbench でコントロールへのデータ入力記録されるとき、記録された値が二重引用符 ("") で囲まれている場合、通常は String データ型であり、String データ型の変数で置き換えることができます。

スクリプトにおける Public および Private 変数

関数および変数は、テスト スクリプトで Public および Private として宣言できます。Private として宣言された変数は、その変数が作成された資産に対してのみ可視化されます。このため、変数がスクリプトまたは関数で作成された場合、その変数はその資産の内部でのみ使用可能になり、資産によって呼び出されるその他の資産では使用できません。

Public として宣言された変数は、作成されたあと、すべての資産で使用可能になります。ある関数から別の関数またはあるスクリプトから別のスクリプトへ渡されるデータはいずれも、Public 変数として宣言する必要があります。ただし、Public 変数を使用すると、Public として宣言されている変数が、同時に再生される別のスクリプトで使用されるという状況が発生する可能性があります。これにより、データが破損したり、コンパイル エラーが生じたりする可能性があります。

変数を Public として宣言するのは、同じ変数名が別の資産で使用されていないことが確実である場合に限定する必要があります。

スクリプト内の Public 変数と Private 変数は、ビジュアル テストでグローバル変数とローカル変数が使用されるのと同じように使用されます。

スクリプトでの変数の宣言

変数は、Dim、Private、または Public ステートメントを使用して宣言できます。構文は以下のとおりです。

```
Dim VariableName As [data type]
```

'or

```
Private VariableName As [data type]
```

'or

```
Public VariableName As [data type]
```

たとえば、String 型の変数を宣言し、その変数に値を割り当てるには、以下のように入力します。

```
Dim carmake As String  
carmake = "Honda"
```

または、以下のコードでは String 型の変数が作成され、その変数に値が割り当てられます。変数が String 型になるのは、「Honda」が String である必要があることがコンパイラによって認識されるためです。このコードの結果は前の例と同じになります。一般的には、以下に示す例をお勧めします。

```
Dim carmake = "Honda"
```

スクリプトで型を指定しないで変数を宣言した場合、Silk Test Workbench によって、変数に対して型が指定されなかったことを示すメッセージが表示されます。デフォルトでは、変数は Object 型であるとコンパイラによって仮定されます。

スクリプト内の変数はすべて宣言することをお勧めします。変数を宣言しない場合、エラーの原因になる場合があります。たとえば、スペルを誤ると 1 つしかないはずの変数が 2 つあると認識され、スクリプトの誤動作の原因になる場合があります。これは非常に一般的なエラーですが、多くの場合その診断は困難です。



注: 手動で変数を強制的に宣言する必要はありません。変数宣言は、デフォルトで強制的に行われます。Option Explicit ステートメントを使用して、スクリプトの変数宣言の強制を手動で無効にすることはできません。これは、スクリプトが起動される前に Silk Test Workbench が非表示の自動インポートを実行するため、Option Explicit ステートメントは、最初のインポートの前に設定する必要があります。

キャプチャされたデータのための変数の作成

オブジェクトの識別 ダイアログ ボックスを使用して、オブジェクト情報をキャプチャします。その後、キャプチャしたロケータ情報を表す変数を作成できます。これを行うには、**オブジェクトの識別** ダイアログ ボックスを使用してオブジェクトをキャプチャし、オブジェクト情報をスクリプトに貼り付けます。以下に示すように、オブジェクトに対して Console.Write メソッドを使用し、オブジェクトに保持されているテキストを取得します。

```
With _desktop.WPFWindow("my main window")
    .WPFFTextBox("@automationId='textBoxSingle']").SetText(carmake)
    Console.Write (carmake)
End With
```

または、以下の例も有効です。

```
<my window variable>.WPFFTextBox("@automationId='textBoxSingle']").SetText(carmake)
Console.Write (carmake)
```

次に、このステートメントの結果、つまりコントロールに保持されるテキストを表す変数を作成できます。データ型に *String* を使用して変数を宣言し、この変数をステートメントの結果に設定します。変数を設定すると、そのデータを評価、検証、表示、または他のテストの目的のために使用することができます。以下は、キャプチャされたテキストの値に設定された変数をメッセージ ボックスに表示します。

```
Dim carmake As String

carmake = window.WPFFTextBox("@automationId='textBoxSingle']").GetLineText(0)

MsgBox (carmake, vbOKOnly)
```

アプリケーションの情報をキャプチャし、変数に設定するこの技法は、テスト アプリケーションで識別可能な任意のオブジェクト（ウィンドウ、リスト ボックス、ボタンなど）で使用することができます。

この目的のためには、Console.Write メソッドを使用する以外に、Text プロパティも使用できます。

コードは以下のようになります。

```
Dim price = browser.DomTextField("@name=txtDealerPrice").Text
```

Text プロパティは、コントロールからのすべてのデータを戻します。ただし、Text プロパティはすべての Silk Test Workbench オブジェクトのメンバではありません。Text プロパティは、サポートされるオブジェクトでデータが表示フィールドのサイズを超える可能性がある場合に使用します。

複数のコントロールまたは画面全体からテキストをキャプチャするには、Console.Write を使用します。たとえば、Web サイトのすべてのテキストを取得するには、以下のように入力します。

```
Console.WriteLine(_desktop.BrowserApplication().Find("//HTML").GetProperty("innerText"))
```

Web サイトのテキストの取得の詳細については、「*textContent*、*innerText*、および *innerHTML* の違い」を参照してください。

変数のデータの検証

再生中にデータ変数の内容を検証する最適な方法は、Visual Basic の MsgBox 機能を使用することです。

再生によって、変数の内容がメッセージ ボックスに表示されます。メッセージ ボックスは、**OK** ボタンをクリックするまで表示されたままになり、クリックすると再生が続行されます。

以下の例は、変数の内容を表示するためのコマンドの簡単な使用例です。

```
Imports SilkTest.Ntf.Wpf

Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        Dim sCarmake = "Honda"
        MsgBox(sCarmake)

        With _desktop.WPFWindow("@caption='WPF Sample Application'")
            .Restore()
            .WPFMenuItem("@caption='Basic Controls']").Select()
        End With
        With _desktop.WPFWindow("@caption='Basic Controls'")
            .WPFTabControl("@automationId='tabControl']").Select("Text")
            .WPFTextBox("@automationId='textBoxSingle']").SetText(sCarmake)
        End With
    End Sub
End Module
```

上記の例では、`sCarmake` 変数値（この場合は `Honda`）が再生中にメッセージボックスに表示されます。



注: この例では、`MsgBox` 機能の実装方法を示すことを意図しています。このコードを正しく実装するには、アプリケーション構成を追加して、テストアプリケーションを必要に応じて調整する必要があります。このサンプルコードをスクリプトに貼り付けても、アプリケーション構成が存在しないため、エラーが発生します。

スクリプトでの変数の使用

任意の Silk Test Workbench コマンド内で二重引用符で囲まれている値（文字列）は、変数の値によって置き換えることができます。通常、これは文字列変数です。ただし、フィールドへの値の入力時に `Integer` や `Double` などの他のデータ型に対して変数を使用できます。

変数の最も一般的な使用法は、エディットボックスなどのコントロールへの入力として使用されるリテラル値を置き換えることです。コントロールにデータを入力する任意のメソッド、たとえば `SetText()` や `TypeKeys()` は、リテラル値を変数で置換できます。そのためには、変数を宣言し、値を割り当て、リテラル値を変数で置き換えます。例：

```
browser.DomTextField("@id='name-search:lastName']").SetText("Smith")
```

'becomes:

```
Dim lastName = "Smith"
```

```
.DomTextField("@id='name-search:lastName']").SetText(sLastName)
```

リストやメニューから項目を選択する場合、入力選択で変数を使用することもできます。例：

```
browser.DomListBox("@id='quick-link:jump-menu']").Select("Agent Lookup")
```

'becomes:

```
Dim selection = "Auto Quote"
```

```
browser.DomListBox("@id='quick-link:jump-menu']").Select(sSelection)
```

メソッドまたはプロパティから返された値を、評価のために変数に入れることができます。この場合の一般的な使用法は、条件ロジックでの評価に対しブールデータ型の値を戻すものです。以下のコードでは、

AllowsMultiSelect プロパティによって戻されたブール型の値を使用して、リスト ボックス コントロールで複数の項目を選択できるかどうかを評価する方法を示しています。

```
Dim canMultiSelect As Boolean

' Returns true if the second list box in the window allows multiple selections
canMultiSelect = mainWindow.ListBox("[2]").AllowsMultiSelect

If canMultiSelect Then
    MsgBox ("Employee selection list box allows to select multiple employees.")
Else
    MsgBox ("Employee selection list box allows to only select one employees.")
EndIf
```

変数は、コントロールの生のアタッチ名で置き換えることができます。文字列連結文字 & を使用します。例：

```
browser.DomLink("@caption='Eye']").Select()

'could also read:

Dim sLink As String
sLink = "Eye"
browser.DomLink("@caption=" & sLink).Select()
```



ヒント: Silk Test Workbench では、GUI マップ、またはコントロールやウィンドウにおける多様性に対処するために、ワイルドカード文字もサポートしています。

スクリプト内のパラメータはビジュアルテストと共有できるため、テストソリューション全体にわたって同じデータを使用できます。詳細については、「スクリプトとビジュアルテスト間でのデータの受け渡し」を参照してください。

スクリプトとビジュアルテスト間でのデータの受け渡し

ビジュアルテストとスクリプトの両方を、同じテストソリューションで使用できます。総合的なテストソリューションを考えた場合、それぞれに利点があります。したがって、テストソリューションには、ビジュアルテストとスクリプトの両方を含めることをお勧めします。スクリプトを使用すると、ビジュアルテストでは実行が困難なタスクを実行できます。また逆に、ビジュアルテストを使用すると、スクリプトでは実行が困難なタスクを実行できます。

テストソリューションへのスクリプトとビジュアルテストの統合を支援するために、Silk Test Workbench ではスクリプトとビジュアルテスト間でデータの受け渡しを行えるようになっています。これにより、既存の Silk Test Workbench スクリプトベースで作成されたデータを再利用できるだけでなく、テストソリューション内でデータの整合性を維持したまま、最も効率的なテスト資産を使用できるようになります。スクリプトとビジュアルテスト間でのデータの受け渡しには、スクリプトパラメータを使用します。



注: スクリプトとビジュアルテスト間でデータの受け渡しを行うには、スクリプトがビジュアルテストに挿入され、そこから再生される必要があります。また、スクリプトとビジュアルテストが同じプロジェクトに含まれているか、挿入されたスクリプトまたはビジュアルテストが参照プロジェクトに含まれている必要があります。


スクリプトパラメータを使用すると、以下のことが可能になります。

- ビジュアルテストでデータを作成し、それをスクリプトで使用する。


- ビジュアル テストでデータを作成し、それをスクリプトに渡して変更し、再びビジュアル テストに戻す。
- スクリプトでデータを作成し、それをビジュアル テストに渡す。

その手順は以下のとおりです。

1. 入力パラメータと出力パラメータをスクリプトに追加します。
2. スクリプトをビジュアル テストに挿入します。
3. ビジュアル テスト内で入力パラメータの値を編集することによって、ビジュアル テストからスクリプトにデータを渡します。

 **注:** ビジュアル テストから送信する値はスクリプト内に保存されませんが、スクリプトでの計算で使用できます。

4. ビジュアル テストで、スクリプトの出力パラメータから渡されたデータを格納するローカル変数を作成し、それをビジュアル テスト内で使用します。

 **注:** 入力パラメータと出力パラメータを作成してビジュアル テストで使用方法の詳細な例については、「*Silk Test Workbench* ビジュアル テスト チュートリアルによるこそ」の「ビジュアル テストからのスクリプトの再生」を参照してください。

ビジュアル テストとスクリプト間で受け渡すパラメータの定義


スクリプトとの間で受け渡されるすべてのパラメータは、スクリプトの入力パラメータおよび出力パラメータとして定義する必要があります。

1. ビジュアル テストとの間でパラメータを受け渡すために使用するスクリプトを開きます。
2. ビジュアル テストからスクリプトに渡されるパラメータを定義する入力パラメータを追加します。
 - a) **プロパティ** ペインで、**入力パラメータの追加** を右クリックして選択します。**スクリプト入力パラメータの追加** ダイアログ ボックスが開きます。
 - b) パラメータの名前を **名前** フィールドに入力します。
 - c) **型** リストからパラメータのデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long Long)	64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

- d) **デフォルト値** テキスト ボックスは空のままにします。
- e) **OK** をクリックします。**プロパティ** ペインの入力パラメータのリストに、入力パラメータが表示されます。

 **注:** スクリプトに対して定義された入力パラメータは、スクリプト自体には表示されません。入力パラメータは **プロパティ** ペインで確認できます。

3. スクリプトからビジュアル テストに渡されるパラメータを定義する出力パラメータを追加します。
 - a) **プロパティ** ペインで、**出力パラメータの追加** を右クリックして選択します。**スクリプト出力パラメータの追加** ダイアログ ボックスが開きます。
 - b) **名前** テキスト ボックスに、ビジュアル テストに渡すパラメータの名前を入力します。

c) **OK** をクリックします。**プロパティ** ペインの出力パラメータのリストに、出力パラメータが表示されます。



注: スクリプトに実際に宣言されている場合を除き、出力パラメータはスクリプトには表示されません。

次のコード サンプルは、渡した変数の使用方法を示します。作成するパラメータを含めるように `Main()` サブルーチンを変更する方法と、ディクショナリを使用してこれらのパラメータにアクセスする方法を示します。入力パラメータとして `inputParameterName` を、出力パラメータとして `outputParameterName` を定義します。

```
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main(args As IDictionary(Of String, Object))
        Dim text As String = args.Item("inputParameterName")
        args.Item("outputParameterName") = "someValue"
    End Sub
End Module
```

4. スクリプトとの間でパラメータを受け渡すために使用するビジュアル テストを開きます。
5. そのスクリプトが少なくとも 1 つのビジュアル テストに含まれている場合、ビジュアル テストで最新のパラメータを表示するには、挿入されたスクリプト ステップの **プロパティ** ペインで、**名前** テキストボックスをクリックし、**パラメータの更新** をクリックします。
6. スクリプトで使用する入力パラメータの値を、ビジュアル テストの **プロパティ** ペインで編集します。ビジュアル テストを再生すると、入力パラメータはスクリプトに渡されます。



注: ビジュアル テストから送信する値はスクリプト内に保存されませんが、スクリプトでの計算で使用できます。

パラメータをビジュアル テストに渡すには、出力パラメータもビジュアル テストでローカル変数として設定する必要があります。

ビジュアル テスト内でのスクリプトの実行

ビジュアル テストでは、既存の .NET スクリプトを呼び出して実行できます。これにより、アプリケーションのテストに対するモジュール式アプローチが可能になり、テストの実行をより制御できるようになります。ビジュアル テスト内でスクリプトを実行する場合は、テスト サイトの整合性を保ち、テスト対象のテスト アプリケーションが正しい初期再生状態になるようにすることが重要です。

この手順を使用してドライバテストを作成することもできます。

1. スクリプトを挿入するビジュアル テストを開きます。
2. 挿入したスクリプトを実行するポイントの前のテスト ステップを選択します。
3. **挿入 > .NET スクリプト** を選択します。**.NET スクリプトの参照** ダイアログ ボックスが開きます。
4. 挿入するスクリプトを選択し、**OK** をクリックします。Silk Test Workbench によって、選択したステップの下にステップが挿入されます。挿入されたステップによって、選択したスクリプトが呼び出され、再生されます。ステップ テキストは、次のようになります：

```
.NET スクリプト '資産名' を再生します
```

ここで、'資産名' は、挿入されたスクリプトの名前です。

再生中、前のステップが実行されると、ビジュアル テスト内の次のステップが実行される前に、挿入されたスクリプトが最後まで再生されます。スクリプトがビジュアル テストに挿入されたあと、スクリプトを編集して、再生に使用される設定を構成できます。

テスト対象アプリケーションは、必要に応じて設定、フック、開始され、再生セッションの終了時までフックされたままになります。たとえば、ビジュアル テストにメモ帳用のアプリケーション構成があり、.NET スクリプトに電卓用のアプリケーション構成がある場合には、次のようになります。

- ビジュアル テストの実行の開始時に、メモ帳がフックされ、開始されます。
- 挿入された .NET スクリプトの実行の開始時に、電卓がフックされ、開始されます。メモ帳はフックされたままです。
- 挿入された .NET スクリプトの実行を停止しても、メモ帳と電卓の両方ともフックされたままになります。
- 再生セッションの終了時、つまり、ビジュアル テストの実行を停止すると、メモ帳と電卓の両方のフックが解除されます。

ビジュアル テストにおける、スクリプトから受け取ったパラメータの使用

スクリプトに対して入力パラメータと出力パラメータが定義されたら、再生中にスクリプトを実行するビジュアル テストに出力パラメータを渡すことができます。ただし、渡されたパラメータをビジュアル テストで使用するには、それをビジュアル テストのローカル変数に設定する必要があります。


1. ビジュアル テストを開きます。
2. **テスト ステップ** ペインで、スクリプトを実行するビジュアル テスト内のステップを選択します。ステップ テキストは、「.NET スクリプト '資産名' を再生します」のようになります (ここで、資産名 は挿入されたスクリプトの名前です)。

ステップのプロパティが **プロパティ** ペインに表示されます。ここには、スクリプトの入力パラメータと出力パラメータが表示されます。

3. ステップの **プロパティ** ペインで、**出力パラメータ** カテゴリを展開します。
スクリプトからビジュアル テストに渡される出力パラメータとして定義された変数の名前が、[出力パラメータ] サブカテゴリに表示されます。これらのパラメータを、ビジュアル テストに作成されたローカル変数に設定する必要があります。


パラメータが **プロパティ** ペインに表示されない場合は、パラメータが含まれるスクリプトを再読み込みします。これを行うには、挿入されたスクリプト ステップの **プロパティ** ペインで、**名前** テキスト ボックスをクリックし、**パラメータ名の更新** をクリックします。

4. **プロパティ** ペインで出力パラメータ名のテキスト ボックスをクリックし、リストから適切なローカル変数を選択します。これにより、スクリプトからビジュアル テストのローカル変数に渡されるパラメータの値が設定されるため、この値をビジュアル テスト内で使用できるようになります。
5. スクリプトからビジュアル テストに渡す **プロパティ** ペインの出力パラメータごとに、上記のステップを繰り返します。

 **注:** スクリプトからビジュアル テストに渡された値を確認するには、**結果コメント** ステップをビジュアル テストに挿入します。コメントの [式] プロパティに対して、**選択** をクリックして **変数** を選択します。渡された値を含むローカル変数をリストから選択します。

6. ビジュアル テストで、適切なテスト ステップを変更してローカル変数を組み込みます。
ローカル変数によって、スクリプトの出力パラメータからテスト ステップにデータが渡されます。
 - a) 使用するステップを **テスト ステップ** ペインで選択します。
 - b) **プロパティ** ペインで、ローカル変数のプロパティ タイプに対応するプロパティのテキスト ボックスをクリックします。
たとえば、**パラメータ** カテゴリの **text** テキスト ボックスをクリックします。
 - c) **選択** ボタンをクリックして、**変数** を選択します。**変数の選択** ダイアログ ボックスが開きます。
 - d) 使用するローカル変数を選択し、**OK** をクリックします。ステップ テキストは、次のようになります：「'[localvariablename]」を入力します」

再生するビジュアル テストにスクリプトが含まれていて、その入力パラメータと出力パラメータが正しく定義されている場合は、ビジュアル テストを再生すると、ビジュアル テストとこのビジュアル テスト内のスクリプトの間でパラメータを正しく受け渡しできます。

 **注:** 入力パラメータと出力パラメータを作成してビジュアル テストで使用方法の詳細な例については、「*SilkTest Workbench* ビジュアル テスト チュートリアルによろこそ」の「ビジュアル テストからのスクリプトの再生」を参照してください。

スクリプトでのパラメータの作成と受け渡し

パラメータを作成して渡すことは、チームによるテスト環境で役立ちます。たとえば、経験の豊富なテスト担当者が共通するテスト機能を実行するスクリプトのライブラリを作成し、経験の浅い開発者がそこからスクリプトを選択して、より基本的なスクリプトを作成することができます。パラメータを作成して渡すことは、1人のテスト担当者の環境でも、複雑な機能を複数のスクリプトで再利用する場合に役立ちます。

親スクリプトと子スクリプト間で、または単一のスクリプトを使用して、パラメータを作成して渡すことができます。親子シナリオを使用することには、複数の子スクリプトで親スクリプトを使用できるという利点があります。たとえば、親スクリプトによって乱数が作成される場合に、その機能を複数の子スクリプトで使用することができます。

スクリプト間でのパラメータの作成と受け渡し

親スクリプトと子スクリプト間で、または単一のスクリプトを使用して、パラメータを作成して渡すことができます。親子シナリオを使用することには、複数の子スクリプトで親スクリプトを使用できるという利点があります。たとえば、親スクリプトによって乱数が作成される場合に、その機能を複数の子スクリプトで使用することができます。

1. 渡すパラメータを組み込む親スクリプトを作成します。

- a) **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
- b) [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。

たとえば、スクリプトに「NameParameters」という名前を付けることができます。

- c) **OK** をクリックして、スクリプトを保存します。Silk Test Workbench によってスクリプトが資産として保存され、スクリプトテンプレートが表示されます。
- d) 作成して子スクリプトに渡すパラメータが含まれるように、Main() サブを変更します。たとえば、以下の親スクリプトは、テストアプリケーションで姓名の文字列を渡します。

```
Public Sub Main()  
    Dim args As New Dictionary(Of String, Object)  
    args("FName") = "Chris"  
    args("LName") = "Smith"
```

- e) 子スクリプトの呼び出しと作成したパラメータの名前指定を行う、RunScript コマンドを組み込みます。たとえば、以下の親スクリプトは、テストアプリケーションで姓名の文字列を渡し、*childscript* という名前の子スクリプトを *args* というパラメータを使用して呼び出します。

```
Public Module Main  
    Dim _desktop As Desktop = Agent.Desktop  
  
    Public Sub Main()  
        Dim args As New Dictionary(Of String, Object)  
        args("FName") = "Chris"  
        args("LName") = "Smith"  
  
        Workbench.RunScript("childscript", args)  
  
    End Sub  
End Module
```

2. 省略可能 : スクリプトで入力されるパラメータを返すメッセージ ボックスを組み込むには、以下のコードを親スクリプトに追加します。

```
MsgBox("Hello, " + args("FName") + "!")  
MsgBox ("Hello, " + args("FName") + " " + args("LName") + "!")
```

スクリプトの再生時に、テストで入力される値を示すメッセージ ボックスが表示されます。

3. 親スクリプトからパラメータを呼び出す、子スクリプトを作成します。
たとえば、サンプル Web アプリケーションの登録フォームで姓名を処理する場合は、姓名を含む登録フォームのスクリプトを記録します。以下に、スクリプトの例を示します。

```
Imports SilkTest.Ntf.XBrowser
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        With _desktop.BrowserWindow("/BrowserApplication[1]//BrowserWindow")
            .DomButton("@id='login-form:signup']").Select()
            .DomTextField("@id='signup:fname']").SetText("first")
            .DomTextField("@id='signup:lname']").SetText("last")
            .DomElement("@src='http://extjs.com/s.gif']").DomClick(MouseButton.Left,
                New Point(8, 16))
            .DomButton("@textContents='Today']").Select()
            .DomTextField("@id='signup:email']").SetText("test@test1.com")
            .DomTextField("@id='signup:street']").SetText("123 street rd")
            .DomTextField("@id='signup:city']").SetText("Marlton")
            .DomListBox("@id='signup:state']").Select("Massachusetts")
            .DomTextField("@id='signup:zip']").SetText("09876")
            .DomTextField("@id='signup:password']").SetText("test")
            .DomButton("@id='signup:signup']").Select()
            .DomButton("@id='signup:continue']").Select()
        End With
    End Sub
End Module
```

4. 設定して親スクリプトに渡す、IDictionary(Of String, Object) という 1 つのパラメータを使用するように Sub Main() を変更します。

IDictionary クラスは、.NET によって提供されます。クラスのドキュメントは、<http://msdn.microsoft.com/en-us/library/s4ys34ea.aspx> で確認できます。

修正された Sub Main () コードは、以下のようになります。

```
Public Sub Main(args As IDictionary(Of String, Object))
```

5. パラメータを使用するコントロールを変更します。

たとえば、Web アプリケーションの記録で姓名を変更するには、以下のコードを置換します。

```
.DomTextField("@id='signup:fname']").SetText("first")
.DomTextField("@id='signup:lname']").SetText("last")
```

以下のように置換します。

```
.DomTextField("@id='signup:fname']").SetText(args("FName"))
.DomTextField("@id='signup:lname']").SetText(args("LName"))
```

6. 親スクリプトから、**再生** をクリックします。



注: 子スクリプトからのパラメータを使用するスクリプトを実行すると、パラメータが指定されていないため、スクリプトは失敗します。ただし、パラメータを使用しない 2 番目の Main() サブを追加し、デフォルトのパラメータ値を他方の Main() に渡して単一のスクリプトから直接実行できます。

単一のスクリプト内でのパラメータの作成と受け渡し

親スクリプトと子スクリプト間で、または単一のスクリプトを使用して、パラメータを作成して渡すことができます。親子シナリオを使用することには、複数の子スクリプトで親スクリプトを使用できるという利点があります。たとえば、親スクリプトによって乱数が作成される場合に、その機能を複数の子スクリプトで使用することがあります。

1. スクリプトを作成します。
 - a) **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
 - b) [資産の種類] リストから **.NET スクリプト** を選択し、**資産名** テキスト ボックスにスクリプトの名前を入力します。

たとえば、スクリプトに「NameParameters」という名前を付けることができます。
 - c) **OK** をクリックして、スクリプトを保存します。Silk Test Workbench によってスクリプトが資産として保存され、スクリプト テンプレートが表示されます。
2. アプリケーションをテストするスクリプトを記録または入力します。
たとえば、サンプル Web アプリケーションの登録フォームで姓名を処理する場合は、姓名を含む登録フォームのスクリプトを記録します。以下に、スクリプトの例を示します。

```
Imports SilkTest.Ntf.XBrowser
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main()
    With _desktop.BrowserWindow("/BrowserApplication[1]//BrowserWindow")
      .DomButton("@id='login-form:signup']").Select()
      .DomTextField("@id='signup:fname']").SetText("first")
      .DomTextField("@id='signup:lname']").SetText("last")
      .DomElement("@src='http://extjs.com/s.gif']").DomClick(MouseButton.Left,
        New Point(8, 16))
      .DomButton("@textContents='Today']").Select()
      .DomTextField("@id='signup:email']").SetText("test@test1.com")
      .DomTextField("@id='signup:street']").SetText("123 street rd")
      .DomTextField("@id='signup:city']").SetText("Marlton")
      .DomListBox("@id='signup:state']").Select("Massachusetts")
      .DomTextField("@id='signup:zip']").SetText("09876")
      .DomTextField("@id='signup:password']").SetText("test")
      .DomButton("@id='signup:signup']").Select()
      .DomButton("@id='signup:continue']").Select()
    End With

  End Sub
End Module
```

3. 作成して子スクリプトに渡すパラメータが含まれる、2 番目の Main() サブを作成します。
たとえば、以下のスクリプトは、テスト アプリケーションで姓名の文字列を渡します。

```
Public Sub Main()
  Dim args As New Dictionary(Of String, Object)
  args("FName") = "Chris"
  args("LName") = "Smith"
```

4. 最初の Main() サブ スクリプトの呼び出しと作成したパラメータの名前指定を行う、Main(args) コマンドを組み込みます。
たとえば、以下のコードは、テスト アプリケーションで姓名の文字列を渡し、args というパラメータを呼び出します。

```
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main()
    Dim args As New Dictionary(Of String, Object)
    args("FName") = "Chris"
    args("LName") = "Smith"

    Main(args)

  End Sub
End Module
```

5. 設定して親スクリプトに渡す、IDictionary(Of String, Object) という 1 つのパラメータを使用するよ
うに、アプリケーションをテストする Sub Main() を変更します。

IDictionary クラスは、.NET によって提供されます。クラスのドキュメントは、<http://msdn.microsoft.com/en-us/library/s4ys34ea.aspx> で確認できます。

修正された Sub Main () コードは、以下のようになります。

```
Public Sub Main(args As IDictionary(Of String, Object))
```

6. パラメータを使用するコントロールを変更します。

たとえば、Web アプリケーションの記録で姓名を変更するには、以下のコードを置換します。

```
.DomTextField("@id='signup:fname").SetText("first")  
.DomTextField("@id='signup:lname").SetText("last")
```

以下のように置換します。

```
.DomTextField("@id='signup:fname").SetText(args("FName"))  
.DomTextField("@id='signup:lname").SetText(args("LName"))
```

スクリプトは以下のようになります。

```
Imports SilkTest.Ntf.XBrowser  
Public Module Main  
    Dim _desktop As Desktop = Agent.Desktop  
  
    Public Sub Main(args As IDictionary(Of String, Object))  
        With _desktop.BrowserWindow("/BrowserApplication[1]//BrowserWindow")  
            .DomButton("@id='login-form:signup").Select()  
            .DomTextField("@id='signup:fname").SetText(args("FName"))  
            .DomTextField("@id='signup:lname").SetText(args("LName"))  
            .DomElement("@src='http://extjs.com/s.gif").DomClick(MouseButton.Left,  
                New Point(8, 16))  
            .DomButton("@textContents='Today').Select()  
            .DomTextField("@id='signup:email").SetText("test@test1.com")  
            .DomTextField("@id='signup:street").SetText("123 street rd")  
            .DomTextField("@id='signup:city").SetText("Marlton")  
            .DomListBox("@id='signup:state").Select("Massachusetts")  
            .DomTextField("@id='signup:zip").SetText("09876")  
            .DomTextField("@id='signup:password").SetText("test")  
            .DomButton("@id='signup:signup").Select()  
            .DomButton("@id='signup:continue").Select()  
        End With  
  
    End Sub  
  
    Public Sub Main()  
        Dim args As New Dictionary(Of String, Object)  
        args("FName") = "Chris"  
        args("LName") = "Smith"  
  
        Main(args)  
  
    End Sub  
  
End Module
```

7. 省略可能 : スクリプトで入力されるパラメータを返すメッセージ ボックスを組み込むには、以下のコー
ドを追加します。

```
MsgBox("Hello, " + args("FName") + "!")  
MsgBox ("Hello, " + args("FName") + " " + args("LName") + "!")
```

スクリプトの再生時に、テストで入力される値を示すメッセージ ボックスが表示されます。

スクリプトは以下のようになります。

```
Imports SilkTest.Ntf.XBrowser
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main(args As IDictionary(Of String, Object))
    With _desktop.BrowserWindow("/BrowserApplication[1]/BrowserWindow")
      .DomButton("@id='login-form:signup']").Select()
      .DomTextField("@id='signup:fname']").SetText(args("FName"))
      .DomTextField("@id='signup:lname']").SetText(args("LName"))
      .DomElement("@src='http://extjs.com/s.gif']").DomClick(MouseButton.Left,
        New Point(8, 16))
      .DomButton("@textContents='Today']").Select()
      .DomTextField("@id='signup:email']").SetText("test@test1.com")
      .DomTextField("@id='signup:street']").SetText("123 street rd")
      .DomTextField("@id='signup:city']").SetText("Marlton")
      .DomListBox("@id='signup:state']").Select("Massachusetts")
      .DomTextField("@id='signup:zip']").SetText("09876")
      .DomTextField("@id='signup:password']").SetText("test")
      .DomButton("@id='signup:signup']").Select()
      .DomButton("@id='signup:continue']").Select()
    End With

    End Sub

    Public Sub Main()
      Dim args As New Dictionary(Of String, Object)
      args("FName") = "Chris"
      args("LName") = "Smith"

      Main(args)

      MsgBox("Hello, " + args("FName") + "!")
      MsgBox ("Hello, " + args("FName") + " " + args("LName") + "!")

    End Sub

  End Module
```

8. 再生 をクリックします。

パラメータを含むスクリプトの例

以下の例は、パラメータを作成して渡すスクリプトを示しています。

リストの例

以下の例は、メモ帳アプリケーションをテストする親スクリプトと子スクリプト間でリストパラメータを作成して渡す方法を示しています。

以下に、親スクリプトの例を示します。

```
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main()
    Dim numList As New List(Of Integer)
    For i As Integer = 0 To 10
      numList.Add(i)
    Next
```



```

Dim args As New Dictionary(Of String, Object)
args("Numbers") = numList

Workbench.RunScript("ChildScriptArgs", args)
End Sub
End Module

```

以下に、子スクリプトの例を示します。

```

Public Module Main
Dim _desktop As Desktop = Agent.Desktop
Public Sub Main(args As IDictionary(Of String, Object))
Dim nums As List(Of Integer) = args("Numbers")

With _desktop.Window("/Window[@caption='Untitled - Notepad']")
For Each num As Integer In nums
.TextField().TypeKeys(num.ToString() + "<Enter>")
Next
End With

End Sub
End Module

```

列挙の例

以下の例は、メモ帳アプリケーションをテストする親スクリプトと子スクリプト間で列挙パラメータを作成して渡す方法を示しています。

以下に、親スクリプトの例を示します。

```

Public Module Main
Dim _desktop As Desktop = Agent.Desktop

Public Sub Main()
Dim args As New Dictionary(Of String, Object)
args("Day") = DayOfWeek.Friday

Workbench.RunScript("ChildDayScript", args)

End Sub
End Module

```

以下に、子スクリプトの例を示します。

```

Public Module Main
Dim _desktop As Desktop = Agent.Desktop

Public Sub Main(args As IDictionary(Of String, Object))
Dim day As DayOfWeek = args("Day")

With _desktop.Window("/Window[@caption='Untitled - Notepad']")
.TextField().TypeKeys("The day of the week is " + day.ToString() + ".")
End With

End Sub
End Module

```

数値の例

以下の例は、メモ帳アプリケーションをテストする単一のスクリプトで数値パラメータを作成して渡す方法を示しています。

スクリプトの内容は以下のとおりです。

```
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main(args As IDictionary(Of String, Object))
    Dim N1 As Integer = CType(args("Num1"), Integer)
    Dim N2 As Integer = CType(args("Num2"), Integer)

    With _desktop.Window("@caption='Untitled - Notepad'")
      .TextField().ClearText()
      .TextField().TypeKeys("<Enter>" + "Numbers" + "<Enter>")
      .TextField().TypeKeys(N1)
      .TextField().TypeKeys("<Enter>")
      .TextField().TypeKeys(N2)
      .TextField().TypeKeys("<Enter>")
    End With
  End Sub

  Public Sub Main()
    Dim args As New Dictionary(Of String, Object)

    args.Add("Num1", 35)
    args.Add("Num2", 25)

    Main(args)
  End Sub
End Module
```

ブール値の例

以下の例は、メモ帳アプリケーションをテストする単一のスクリプトでブール値パラメータを作成して渡す方法を示しています。

スクリプトの内容は以下のとおりです。

```
Public Module Main
  Dim _desktop As Desktop = Agent.Desktop

  Public Sub Main(args As IDictionary(Of String, Object))
    Dim B1 As Boolean = CType(args("Bool1"), Boolean)
    Dim B2 As Boolean = CType(args("Bool2"), Boolean)

    With _desktop.Window("@caption='Untitled - Notepad'")
      .TextField().ClearText()
      .TextField().TypeKeys("<Enter>" + "Boolean Values" + "<Enter>")
      .TextField().TypeKeys(B1)
      .TextField().TypeKeys("<Enter>")
      .TextField().TypeKeys(B2)
      .TextField().TypeKeys("<Enter>")
    End With
  End Sub

  Public Sub Main()
    Dim args As New Dictionary(Of String, Object)

    args.Add("Bool1", True)
```

```
args.Add("Bool2", False)

Main(args)
End Sub
End Module
```

配列の例

以下の例は、メモ帳アプリケーションをテストする単一のスクリプトにおいて、配列パラメータを作成して渡す方法を示しています。

スクリプトの内容は以下のとおりです。

```
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main(args As IDictionary(Of String, Object))
        Dim A1() As Integer = CType(args("Arr1"), Integer())

        With _desktop.Window("@caption='Untitled - Notepad'")
            .TextField().TypeKeys("Array Numbers<Enter>")

            For Each num As Integer In A1
                .TextField().TypeKeys(num)
                .TextField().TypeKeys("<Enter>")
            Next
            .TextField().TypeKeys("<Enter>")

        End With
    End Sub

    Public Sub Main()
        Dim args As New Dictionary(Of String, Object)

        args("Arr1") = New Integer() {1,2,3,4,5}

        Main(args)
    End Sub
End Module
```

スクリプトの管理

このセクションでは、VB .NET スクリプトの管理方法について説明します。

スクリプト内のコードの検索と置換

VB .NET スクリプト内のコードの検索と置換を使用すると、スクリプトを迅速に変更できます。

1. 検索するスクリプトを開きます。
2. **編集 > 検索と置換** を選択し、以下のいずれかのオプションを選択します。
 - **検索** : **検索する文字列** テキスト ボックスに検索するコードを指定し、**次を検索** をクリックします。
 - **次を検索** : **検索する文字列** テキスト ボックスに指定した項目の次のインスタンスを検索するには、このオプションを選択します。
 - **前を検索** : 検索された最後のエントリの前にあるスクリプトで、**検索する文字列** テキスト ボックスに指定した項目を検索するには、このオプションを選択します。
 - **置換** : **検索する文字列** テキスト ボックスに検索するコードを指定します。**置換後の文字列** テキスト ボックスに、既存のコードと置換するコードを指定し、**置換** または **すべて置換** をクリックします。

ブックマークを使用したスクリプトでのコードの移動

ブックマークを使用して、スクリプトの重要な部分に移動します。

1. ブックマークを組み込むスクリプトを開きます。
2. **編集 > ブックマーク** を選択し、以下のいずれかのオプションを選択します。
 - **ブックマークの切り替え**: 選択したコード行でブックマークを組み込むか非表示にするには、このオプションを選択します。
 - **次のブックマーク**: スクリプト内の次のブックマークに移動するには、このオプションを選択します。
 - **前のブックマーク**: スクリプト内の検索された最後のブックマークの前のブックマークに移動するには、このオプションを選択します。
 - **ブックマークのクリア**: スクリプトからすべてのブックマークを削除するには、このオプションを選択します。

スクリプトの入力パラメータの追加

スクリプトは、入力パラメータを使用してビジュアル テストからデータを受け取ったり、逆に出力パラメータを使用してビジュアル テストにデータを渡したりできます。

1. **プロパティ ペイン**で、**入力パラメータの追加** を右クリックして選択します。 **スクリプト入力パラメータの追加** ダイアログ ボックスが開きます。

2. パラメータの名前を **名前** フィールドに入力します。

パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。

3. **型** リストからパラメータのデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long Long)	64 ビット整数値(-9223372036854775808 から 9223372036854775807 の範囲) です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

4. **デフォルト値** テキスト ボックスは空のままにします。

空のままにした場合、ビジュアル テストで設定されたパラメータの初期値が使用されます。この値は、ビジュアル テスト、またはスクリプトに渡された場合はスクリプトで変更できます。値は型と整合性がある必要があります。たとえば、型が数値 (Long) の場合、初期値は整数にする必要があります。

5. **OK** をクリックします。 **プロパティ ペイン**の入力パラメータのリストに、入力パラメータが表示されます。

スクリプトの入力パラメータの編集

スクリプトの入力パラメータを編集して、パラメータを渡すスクリプトやビジュアル テストに行える変更を反映します。

1. **プロパティ** ペインで、編集する入力パラメータを右クリックし、**<inputparametername> の編集** を選択します。**スクリプト入力パラメータの編集** ダイアログ ボックスが開きます。
2. パラメータの名前を **名前** フィールドに入力します。
パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。
3. **型** リストからパラメータのデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long Long)	64 ビット整数値(-9223372036854775808 から 9223372036854775807 の範囲) です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

4. **デフォルト値** テキスト ボックスは空のままにします。
空のままにした場合、ビジュアル テストで設定されたパラメータの初期値が使用されます。この値は、ビジュアル テスト、またはスクリプトに渡された場合はスクリプトで変更できます。値は型と整合性がある必要があります。たとえば、型が数値 (Long) の場合、初期値は整数にする必要があります。
5. **OK** をクリックします。

スクリプトからの入力パラメータの削除

必要に応じて、スクリプトから入力パラメータを削除できます。

1. **プロパティ** ペインで、削除する入力パラメータを右クリックし、**<inputparametername> の削除** を選択します。
2. **はい** をクリックして、入力パラメータを削除することを確認します。

スクリプトの出カパラメータの追加

スクリプトは、入力パラメータを使用してビジュアル テストからデータを受け取ったり、逆に出力パラメータを使用してビジュアル テストにデータを渡したりできます。

1. **プロパティ** ペインで、**出力パラメータの追加** を右クリックして選択します。**スクリプト出力パラメータの追加** ダイアログ ボックスが開きます。
2. **名前** テキスト ボックスに、ビジュアル テストに渡すパラメータの名前を入力します。
パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。
3. **型** リストから、データを渡すビジュアル テスト内のローカル変数のデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。

数値 (Long Long)	64 ビット整数値(-9223372036854775808 から 9223372036854775807 の範囲) です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

4. **OK** をクリックします。 **プロパティ** ペインの出力パラメータのリストに、出力パラメータが表示されます。

スクリプトの出力パラメータの編集

ビジュアルテストにデータを渡すために、スクリプトの出力パラメータを編集します。

1. **プロパティ** ペインで、編集する出力パラメータを右クリックし、 **<outputparametername> の編集** を選択します。
2. **名前** テキスト ボックスに、ビジュアルテストに渡すパラメータの名前を入力します。
パラメータ名に有効な文字は、大文字/小文字の英数字とアンダースコア (「ABC」、「abc」、「_」) です。追加または編集するパラメータ名が他のパラメータ名と重複している場合、そのパラメータに対して一意な名前が付けられます。
3. **型** リストから、データを渡すビジュアルテスト内のローカル変数のデータ型を選択します。

Silk Test Workbench は、次のパラメータの型をサポートします。

ブール値	パラメータの値は True または False です。
数値 (Double)	パラメータの値は 64 ビット数値 (-1.7E308 から +1.7E-307 の範囲) として格納された倍精度浮動小数点数です。
数値 (Long Long)	64 ビット整数値(-9223372036854775808 から 9223372036854775807 の範囲) です。
数値 (Long)	パラメータの値は -2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。
テキスト	パラメータの値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

4. **OK** をクリックします。

スクリプトからの出力パラメータの削除

必要に応じて、スクリプトから出力パラメータを削除できます。

1. **プロパティ** ペインで、削除する出力パラメータを右クリックし、 **<outputparametername> の削除** を選択します。
2. **はい** をクリックして、出力パラメータを削除することを確認します。

スクリプトへのファイルの追加

スクリプトが参照するファイルを追加して、特定の目的ごとにコードを論理的にセグメント化できます。たとえば、スクリプトで 4000 行のコードが必要な場合、 **メイン** ウィンドウにはコードの中心的な行のみを含めるために、ヘルパコード用の別のファイルを作成することがあります。ヘルパコードとは、データ転送のためにメイン アプリケーションをデータベースに接続するコードなどのことです。

1. ファイルをスクリプトに追加するには、以下のいずれかのステップを実行します。

- **挿入 > ファイル** を選択します。
- **プロパティ** ペインを右クリックし、**ファイルの追加** を選択します。

コード ウィンドウで新しいタブが開きます。

2. ファイルに含めるコードを追加し、**保存** をクリックします。
3. **メイン** タブをクリックし、ファイルの参照を追加します。

スクリプトからのファイルの削除

スクリプトが参照するファイルを削除できます。

1. 削除するファイルが含まれているスクリプトを開きます。
2. **プロパティ** ペインで、**ファイル** カテゴリを展開し、ファイル名を右クリックして **ファイルの削除** を選択します。

スクリプト内のファイル名の変更

スクリプトが参照するファイルの名前を変更できます。

1. 名前を変更するファイルが含まれているスクリプトを開きます。
2. **プロパティ** ペインで、**ファイル** カテゴリを展開し、ファイル名を右クリックして **ファイル名の変更** を選択します。
3. **プロパティ** ペインで新しい名前を入力します。

.NET スクリプト内からの.NET アセンブリの参照

スクリプトを記録する場合、操作するコントロールに対して適切なすべての参照が参照ライブラリに自動的に追加されます。手動でスクリプトを作成するか、またはアプリケーション構成を追加すると、参照を手動でライブラリに追加することが必要となる場合があります。.NET スクリプトの参照として、.NET Framework の .NET アセンブリを追加できます。

1. スクリプト参照ライブラリに参照を追加するには、**プロパティ** ペインを右クリックし、**.NET アセンブリ参照の追加** を選択します。**.NET アセンブリ参照の追加** ダイアログ ボックスが開きます。
2. コンポーネントを追加するには、以下のいずれかのステップを実行します。
 - コンポーネント リストからコンポーネントをダブルクリックします。
 - **参照** をクリックし、追加するコンポーネントに移動し、**開く** をクリックして、**OK** をクリックします。

パス 列には、DLL ファイルの場所の一覧が表示されます。

新しいコンポーネントが**プロパティ** ペインの **参照** カテゴリに表示されます。

.NET スクリプト内から他の .NET スクリプトのクラスの参照

.NET スクリプトで他の .NET スクリプトで定義したクラスを使用するには、他の .NET スクリプトへの参照を .NET スクリプトに追加します。

1. **プロパティ** ペインを右クリックし、**.NET スクリプト参照の追加** を選択します。**.NET スクリプトの参照** ダイアログ ボックスが開きます。
2. ダイアログ ボックスに表示する .NET スクリプトを選択します。
 - 自分で作成したすべての .NET スクリプトを表示する場合は、**自分で作成したファイル** をクリックします。
 - すべてのユーザーが作成したすべての .NET スクリプトを表示する場合は、**すべてのユーザーが作成したファイル** をクリックします。
3. リストから .NET スクリプトを参照として現在の .NET スクリプトに追加するには、以下のステップのいずれかを実行します。

- リストから .NET スクリプトをダブルクリックします。
- リストから .NET スクリプトを選択して、**OK** をクリックします。
- リストから .NET スクリプトを選択して、**挿入して開く** をクリックして、.NET スクリプトを参照として追加して、参照したスクリプトを開きます。


参照した .NET スクリプトは **プロパティ** ペインの **参照** カテゴリに表示されます。

スクリプト参照ライブラリからの参照の削除

スクリプト参照ライブラリから参照を削除するには、**プロパティ** ペインを右クリックし、**参照の削除** を選択します。

[.NET スクリプトの参照] ダイアログ ボックス


.NET スクリプトの参照 ダイアログ ボックスを使用して、自分自身または他のユーザーが作成したスクリプトをすばやく検索して選択できます。

 **注:** **資産の選択** リストに表示されるのは、*Common* プロジェクトのスクリプトのみです。

.NET スクリプトの参照 ダイアログ ボックスには、以下のコントロールがあります。

コントロール	説明
作成した資産の表示	資産の選択 リストに表示する資産を定義するためのリストボックスです。自分で作成したファイルを選択すると、ユーザーが作成した資産のみが表示され、すべてのユーザーが作成したファイルを選択すると、任意のユーザーが作成した資産が表示されます。
名前フィルタ	資産の名前をこのテキスト フィールドに入力すると、 資産の選択 リストにその名前と一致した資産のみが表示されます。
資産の選択	適用したフィルタによって表示が変化する資産のリストです。


資産の選択 リストから目的の VB .NET スクリプトを選択し、**OK** ボタンを選択して使用します。

 **注:** 列ヘッダーをクリックすると、クリックした列によって **資産の選択** リストがアルファベット順にソートされます。

.NET スクリプトの比較

2つの異なる .NET スクリプトや、同じ .NET スクリプトの2つのバージョンを比較して、これらの間の相違点を調べることができます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面** の **入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **.NET スクリプト** を選択します。選択したプロジェクトの既存の .NET スクリプトが、右側のペインに表示されます。

 **ヒント:** 調査したい .NET スクリプトが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。


3. 2つの異なる .NET スクリプトを比較するには :
 - a) **Ctrl + クリック** を使用して、比較する2つの .NET スクリプトを選択します。
 - b) 選択した項目を右クリックします。
 - c) **相違点の表示** をクリックします。
4. 同じ .NET スクリプトの2つのバージョンを比較するには :
 - a) 調査したい .NET スクリプトを右クリックして、**すべてのバージョンを表示** を選択します。
 - b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する2つのバージョンを選択します。
 - c) **相違点の表示** をクリックします。

- 変更箇所は黄色で表示されます。
- 追加箇所は緑色で表示されます。
- 比較した資産の一方に存在しない項目はグレーで表示されます。


.NET スクリプトのマージ

2つの異なる .NET スクリプトや、同じ .NET スクリプトの2つのバージョンを比較して、.NET スクリプト間、または .NET スクリプトのバージョン間の相違点をマージすることができます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **.NET スクリプト** を選択します。選択したプロジェクトの既存の .NET スクリプトが、右側のペインに表示されます。

 **ヒント:** 調査したい .NET スクリプトが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 2つの異なる .NET スクリプトを比較するには：
 - a) **Ctrl + クリック** を使用して、比較する2つの .NET スクリプトを選択します。
 - b) 選択した項目を右クリックします。
 - c) **相違点の表示** をクリックします。
 4. 同じ .NET スクリプトの2つのバージョンを比較するには：
 - a) 調査したい .NET スクリプトを右クリックして、**すべてのバージョンを表示** を選択します。
 - b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する2つのバージョンを選択します。
 - c) **相違点の表示** をクリックします。
 5. 相違点の表示を移動するには、ツールバーの **次の相違点** と **前の相違点** を使用します。
 6. 選択した相違点をマージするには：
 - **左側にコピー** をクリックすると、右側のペインに表示されているバージョンまたは資産の選択した相違点を、左側に表示されているバージョンまたは資産にマージします。
 - **右側にコピー** をクリックすると、左側のペインに表示されているバージョンまたは資産の選択した相違点を、右側に表示されているバージョンまたは資産にマージします。
- 変更箇所は黄色で表示されます。
 - 追加箇所は緑色で表示されます。
 - 比較した資産の一方に存在しない項目はグレーで表示されます。


 **注:** Silk Test Workbench ユーザーが資産のアクティブバージョンを比較またはマージしている間、その資産はロックされ、他のユーザーは編集できなくなります。資産が他のウィンドウで編集用に既に開かれている場合に、その資産のアクティブバージョンを比較またはマージしようとすると、読み取り専用モードでそのバージョンが開かれます。読み取り専用かどうかは、そのバージョンのタイトルバーに表示されます。


Windows DLL の呼び出し


このセクションでは、DLL を呼び出す方法について説明します。DLL は Open Agent のプロセス内から、または AUT (テスト対象アプリケーション) から呼び出すことができます。これにより、テスト スクリプト内の既存のネイティブ DLL を再利用できます。

Open Agent 内の DLL 呼び出しは通常、AUT 内の UI コントロールと対話しないグローバル関数を呼び出す場合に使用されます。

AUT 内の DLL 呼び出しは通常、アプリケーションの UI コントロールと対話する関数を呼び出す場合に使用されます。これにより、Silk Test Workbench は再生中に DLL 呼び出しを自動的に同期できます。

 **注:** 32 ビット アプリケーションでは 32 ビット DLL を、64 ビット アプリケーションでは 64 ビット DLL を呼び出すことができます。Open Agent は 32 ビットと 64 ビットの両方の DLL を実行できます。

 **注:** .NET Framework では、P/Invoke という DLL 呼び出しも組み込みでサポートされています。P/Invoke を Visual Basic スクリプト内で使用すると、このスクリプトを実行するプロセス内で DLL 関数を呼び出すことができます。ただし、AUT では Silk Test Workbench を使用して DLL 関数を呼び出すことができる一方で、自動同期は行われません。

 **注:** DLL を呼び出すには、C インターフェイスを使用する必要があります。同様に .dll というファイル拡張子の付いた .NET アセンブリを呼び出す場合は、DLL 呼び出し機能を使用しないで、.NET スクリプト内でアセンブリへの参照を追加します。

スクリプトからの Windows DLL の呼び出し

DLL の宣言を開始するには、DLL 属性を持つインターフェイスを使用します。宣言の構文は次のとおりです。

```
<Dll("dllname.dll")> Public Interface DllInterfaceName
    FunctionDeclaration
    [FunctionDeclaration]...
End Interface
```

dllname Visual Basic スクリプトから呼び出す関数が含まれた DLL ファイルの完全パスの名前。DLL パスの環境変数は自動的に解決されます。パスでは二重のバックスラッシュ (¥¥) を使用する必要はありません。単一のバックスラッシュ (¥) で十分です。

DllInterfaceName スクリプト内で DLL と対話するために使用される識別子。

FunctionDeclaration 呼び出そうとしている DLL 関数の関数宣言。

DLL 関数の宣言構文

DLL 関数の宣言は、一般に以下の形式を取ります。

```
Function function-name( [arg-list] ) As return-type
```

戻り値のない関数の場合、宣言の形式は以下のとおりです、

```
Sub function-name( [arg-list] )
```

return-type 戻り値のデータ型。

function-name 関数の名前。

arg-list 関数に渡される引数のリスト。
リストは以下のように指定します。

```
[pass-mode] identifier As data-type
```

pass-mode 引数が関数に渡されるのか (ByVal)、それとも引数を関数によって変更できるのか (ByRef) を指定します。ByRef キーワードを使用して、引数の値を関数に渡すこともできます。この場合、関数によって引数の値が変更され、新しい値が出力されます。

data-type 引数のデータ型。

identifier 引数の名前。

DLL 呼び出しの例


この例では、user32.dll の SendMessage DLL 関数を呼び出して、フィールドに「hello world!」というテキストを書き出します。

DLL の宣言：

```
// VB .NET code
<Dll("user32.dll")> Public Interface IUserDll32Functions
    Function SendMessageW( _
        ByVal obj As TestObject, ByVal message As Integer, ByVal wParam As Integer, ByVal lParam As String) As Integer
End Interface
```


以下のコードは、AUT で宣言された DLL 関数を呼び出す方法を示します。

```
// VB .NET code
Public Sub Main()
    Dim user32Functions As IUserDll32Functions = DllCall.CreateInProcessDllCall(Of IUserDll32Functions)()
    Dim textField = _desktop.Window().TextField()
    user32Functions.SendMessageW(textField, WindowsMessages.WM_SETTEXT, 0, "my text")
End Sub
```

 **注:** DLL 関数の最初のパラメーターに C データ型の HWND が指定されている場合は、AUT 内で DLL 関数の呼び出しのみを実行できます。

次のコードは、Open Agent のプロセスで宣言された DLL 関数を呼び出す方法を示します。

```
// VB .NET code
Public Sub Main()
    Dim user32Functions As IUserDll32Functions = DllCall.CreateAgentDllCall(Of IUserDll32Functions)()
    Dim textField = _desktop.Window().TextField()
    user32Functions.SendMessageW(textField, WindowsMessages.WM_SETTEXT, 0, "my text")
End Sub
```

 **注:** コード例では、DLL 関数で使用するのに便利な Windows メッセージングに関連する定数を定義した WindowsMessages クラスを使用しています。

DLL 関数への引数の受け渡し




DLL 関数は C で記述されているため、これらの関数に渡す引数には適切な C データ型を指定する必要があります。次のデータ型がサポートされます。

Integer

次のデータ型を持つ引数または戻り値には、このデータ型を使用します。

- int
- INT
- long
- LONG
- DWORD
- BOOL
- WPARAM
- HWND

Visual Basic の Integer 型は、4 バイト値を持つすべての DLL 引数に対して有効です。

Long	C データ型 long および int64 を持つ引数または戻り値には、このデータ型を使用します。Visual Basic の Long 型は、8 バイト値を持つすべての DLL 引数に対して有効です。
Short	C データ型 short および WORD を持つ引数または戻り値には、このデータ型を使用します。Visual Basic の Short 型は、2 バイト値を持つすべての DLL 引数に対して有効です。
Boolean	C データ型 bool を持つ引数または戻り値には、このデータ型を使用します。
String	C で String となる引数または戻り値には、このデータ型を使用します。
Double	C データ型 double を持つ引数または戻り値には、このデータ型を使用します。
Single	C データ型 float を持つ引数または戻り値には、このデータ型を使用します。
SilkTest.Ntf.Rectangle	C データ型 RECT を持つ引数には、このデータ型を使用します。Rectangle は戻り値として使用できません。
SilkTest.Ntf.Point	C データ型 POINT を持つ引数には、このデータ型を使用します。POINT は戻り値として使用できません。
SilkTest.Ntf.TestObject	C データ型 HWND を持つ引数には、このデータ型を使用します。TestObject は戻り値として使用できませんが、戻り値型として Integer を持つ HWND を戻す DLL 関数を宣言できます。
	 注: 渡された TestObject は SilkTest.Ntf.INativeWindow インターフェイスを実装して、DLL 関数に渡される TestObject のウィンドウハンドルを Silk Test Workbench が判別できるようにする必要があります。そうしないと、この DLL 関数を呼び出すときに、例外がスローされます。
List	ユーザー定義の C 構造体の配列には、このデータ型を使用します。List は戻り値として使用できません。
	 注: List を出力パラメーターとして使用する場合は、ByRef を使用して、渡されるリストに、戻される内容を保持できるだけのサイズを確保する必要があります。
	 注: C 構造体は List で表すことができます。この場合、すべてのリスト要素は構造体のメンバに対応しています。最初の構造体メンバは、リスト内の最初の要素で表されます。2 番目の構造体メンバは、リスト内の 2 番目の要素で表されます (以下同様)。

 **注:** DLL 関数に渡す引数の前には、いずれかの Visual Basic データ型を配置する必要があります。

DLL 関数で変更できる引数の受け渡し

値が DLL 関数によって変更される引数は、ByRef キーワード を使用して宣言する必要があります。

例

この例では、現在のカーソル位置を取得するために、user32.dll の GetCursorPos 関数を使用しています。

DLL の宣言 :

```
<Dll( "user32.dll" )> Public Interface IUserDll32Functions
    Function GetCursorPos( ByRef point As Point) As Integer
End Interface
```

使用法 :

```
Public Sub Main()  
    Dim user32Functions As IUserDll32Functions =  
    DllCall.CreateAgentDllCall( Of IUserDll32Functions )()  
    Dim point As Point  
    user32Functions.GetCursorPos(point)  
    MsgBox( "cursor position = (" & point.X & ", " & point.Y & ")" )  
End Sub
```

DLL 関数への文字列引数の受け渡し

DLL 関数に渡している文字列、または DLL 関数から戻される文字列は、デフォルトでは Unicode Strings として処理されます。DLL 関数に ANSI String 引数が必要な場合は、DllFunctionOptions 属性の CharacterSet プロパティを使用します。

例

```
<Dll( "user32.dll" )> Public Interface IUserDll32Functions  
    <DllFunctionOptions(CharacterSet:=CharacterSet.Ansi)> Function  
    SendMessageA( _  
        ByVal obj As TestObject, ByVal message As Integer , ByVal wParam  
        As Integer , ByRef lParam As String ) As Integer  
End Interface
```

DLL 呼び出しから String を ByRef 引数として戻した場合、String のサイズが 256 文字以下であれば、デフォルトの動作に従います。戻される String が 256 文字を超えている場合は、作成された String を保持できるだけの長さを持つ、Visual Basic String を渡します。

例

```
1024 個の空白文字を含む String を作成するには、以下のコードを使用します。  
Dim longEmptyString = New String ( " "c , 1024 )
```

この String を ByRef 引数として DLL 関数に渡します。すると、この DLL 関数は最大 1024 文字の String を戻します。

関数の戻り値として DLL から String が戻される場合、DLL は DLL 関数 FreeDllMemory を実装し、DLL 関数から戻される C String ポインターを受け入れて、以前に割り当てられたメモリーを解放する必要があります。このような関数が存在しない場合、メモリーはリークされます。

DLL 名のエイリアス設定

DLL 関数に、Visual Basic の予約語と同じ名前が付いている場合、または DLL 関数に名前ではなく序数が付いている場合は、宣言内でこの関数の名前を変更し、エイリアス ステートメントを使用して、宣言した名前と実際の名前をマッピングする必要があります。

例

たとえば、Exit ステートメントは Visual Basic コンパイラーで予約されています。したがって、関数 exit を呼び出すには、次のようにその関数を別の名前で宣言し、エイリアス ステートメントを追加する必要があります。

```
<Dll("mydll.dll")> Public Interface IMyDllFunctions
    <DllFunctionOptions(Alias:="exit")> Sub MyExit()
End Interface
```

DLL 関数呼び出しの表記規則

DLL 関数を呼び出す場合は、次に示す呼び出し規則がサポートされています。

- `__stdcall`
- `__cdecl`

DLL 関数を呼び出す場合は、`__stdcall` 呼び出し規則がデフォルトで使用されます。この呼び出し規則は、すべての Windows API DLL 関数で使用されます。

DLL 関数の呼び出し規則を変更するには、`DllFunctionOptions` 属性の `CallingConvention` プロパティを使用します。

例

次のコード例では、`__cdecl` 呼び出し規則を使用して DLL 関数を宣言します。

```
<Dll("msvcrt.dll")> Public Interface IMsVisualCRuntime
    <DllFunctionOptions(CallingConvention:=CallingConvention.Cdecl)>
    Function cos(ByVal input As Double) As Double
End Interface
```

Microsoft ユーザー補助を使用したオブジェクト解決の向上

Microsoft ユーザー補助を、クラス レベルでオブジェクトを簡単に認識するために使用することができます。Internet Explorer や Microsoft アプリケーションのいくつかのオブジェクトには、ユーザー補助を有効にすることで Silk Test Workbench によってより良く認識されるようになります。たとえば、ユーザー補助を有効にしないと、Silk Test Workbench は Microsoft Word のメニュー バーや表示されるタブについて基本的な情報のみを記録します。しかし、ユーザー補助を有効にすると、Silk Test Workbench はそれらのオブジェクトを完全に認識するようになります。

例

ユーザー補助を使用しないと、Silk Test Workbench は `DirectUIHwnd` コントロールを完全に認識できません。これは、このコントロールのパブリックな情報が存在しないためです。Internet Explorer は、2 つの `DirectUIHwnd` コントロールを使用しています。1 つはブラウザ ウィンドウの下部に表示されるポップアップです。このポップアップには、通常、次の情報が表示されます。

- Internet Explorer を既定のブラウザにしたいかどうかを尋ねるダイアログ ボックス。
- ダウンロード オプション（開く、保存、キャンセル）。

Silk Test Workbench でプロジェクトを開始して、`DirectUIHwnd` ポップアップに対してロケータを記録すると、ユーザー補助を無効にしている場合、単一のコントロールのみが表示されます。ユーザー補助を有効にした場合には、`DirectUIHwnd` コントロールを完全に認識した情報が得られます。

ユーザー補助の使用

Win32 では、ジェネリック コントロールとして認識されるコントロールにユーザー補助 サポートが使用されます。Win32 は、コントロールを特定すると、ユーザー補助オブジェクトをコントロールのすべてのユーザー補助の子とともに取得しようとします。

ユーザー補助によって返されるオブジェクトは、AccessibleControl、Button、CheckBox のいずれかのクラスになります。Button および Checkbox は、そのクラス用に定義されたメソッドとプロパティの通常セットをサポートするので個別に扱われます。ユーザー補助によって返されるすべてのジェネリック オブジェクトの場合、クラスは AccessibleControl です。

例

ユーザー補助が有効になる前、アプリケーションのコントロール階層が次のようになっていたとします。

- コントロール
 - コントロール
- ボタン

ユーザー補助を有効にすると、階層は次のようになります。

- コントロール
 - コントロール
 - ユーザー補助コントロール
 - ユーザー補助コントロール
 - ボタン
- ボタン

ユーザー補助の有効化

Win32 アプリケーションをテストしているときに、がオブジェクトを認識できない場合は、最初にユーザー補助を有効にする必要があります。ユーザー補助は、オブジェクトの認識機能をクラス レベルで強化するためのものです。

ユーザー補助を有効にするには、以下の手順を実行します。

1. ツール > オプション をクリックします。オプション ダイアログ ボックスが表示されます。
2. 詳細設定 をクリックします。
3. Microsoft ユーザー補助を使用する オプションを選択します。ユーザー補助が有効になります。

Microsoft UI オートメーション

Microsoft UI オートメーション (UI オートメーション) は、アプリケーションの UI 要素をプログラム上からアクセス、識別、操作できるようにするためのフレームワークです。UI オートメーション プロバイダー インターフェイスを実装する Windows ベースのアプリケーションに対してテストを実行する場合、UI オートメーションを使うと、アプリケーションのコントロールに対するオブジェクト解決を改善することができます。このようなコントロールのことを、以降では UI オートメーション コントロール と呼びます。




注: Silk Test Workbench は、UI オートメーションを実装した Windows ベースのアプリケーションに対する Microsoft Windows 8 以降のマシン上でのテストをサポートします。


UI オートメーションは、次のテクノロジーを利用したアプリケーションに対するフォールバック サポートとして利用できます。

- Win32
- WPF
- WinForms
- Oracle JavaFX
- QT
- PowerBuilder
- Delphi
- Microsoft Office

たとえば、Silk Test Workbench がアプリケーションのオブジェクトを解決できなかつたり、Silk Test Workbench がアプリケーションのすべてのオブジェクトを Control として解決することが原因で、アプリケーションに対するテストを記録できない場合、UI オートメーション サポートを有効化してみることができま

す。UI オートメーション サポートを記録時に有効にするには、記録を停止して **Microsoft UI オートメーション オプション** を有効にしてから、記録を再開します。詳細については、「*UI オートメーション オプションの設定*」を参照してください。

 **注:** UI オートメーション サポートは、標準のテクノロジー ドメイン固有のサポートをオーバーライドします。UI オートメーション サポートが必要なコントロールに対する操作を完了したら、UI オートメーション サポートをオフにして、標準コントロールに対する操作に戻すこともできます。

 **注:** Java FX アプリケーションをテストする場合は、UI オートメーション サポートをオンにする必要はありません。Silk Test Workbench では、Java FX アプリケーションに対して、デフォルトでオンになります。

UI オートメーション コントロールのサポートするメソッドが、すべてのサポートするテクノロジーの対応するコントロールに対して有効であることを保障するため、Silk Test Workbench API がサポートするのは、これらのコントロールに対して利用できるメソッドやプロパティのサブセットになります。コントロールに対して Silk Test Workbench API で利用できないその他のメソッドやプロパティの呼び出しには、動的呼び出しを使用してください。

UI オートメーションプロバイダー インターフェイスを実装したアプリケーションに対するテストの記録

インタラクティブな **記録** ウィンドウを使って、UI オートメーションプロバイダー インターフェイスを実装した Windows ベースのアプリケーションに対してテストを記録できます。

 **注:** Silk Test Workbench は、UI オートメーションを実装した Windows ベースのアプリケーションに対する Microsoft Windows 8 以降のマシン上でのテストをサポートします。

UI オートメーションプロバイダー インターフェイスを実装した Windows ベースのアプリケーションに対して新しいテストを記録するには：

1. **ファイル > 新規作成** を選択します。**資産の新規作成** ダイアログ ボックスが開きます。
2. 記録するテスト タイプを選択します。
 - 新しい .NET スクリプトを記録する場合は、資産の種類リストから **.NET スクリプト** を選択します。
 - 新しいビジュアル テストを記録する場合は、資産の種類リストから **ビジュアル テスト** を選択します。
3. 新しいテストの名前を **資産名** フィールドに入力します。
4. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
5. **OK** をクリックします。**アプリケーションの選択** ダイアログが表示されます。
6. **Windows** タブを選択します。


7. テストするアプリケーションを選択します。
 8. **OK** をクリックします。
 9. インタラクティブな **記録** ウィンドウが開き、テスト対象アプリケーションを表示します。記録したい操作を実行します。
 - a) 操作したいオブジェクトをクリックします。Silk Test Workbench は、オブジェクトのデフォルトの操作を実行します。デフォルトの操作がない場合や、テキストを挿入したりパラメータを指定する場合は、**操作の選択** ダイアログ ボックスが開きます。
 - b) 省略可能：デフォルトの操作以外のオブジェクトの操作を選択するには、オブジェクトを右クリックします。**操作の選択** ダイアログ ボックスが開きます。
 - c) 省略可能：操作にパラメータある場合は、パラメータ フィールドにパラメータを入力します。Silk Test Workbench は自動的にパラメータを検証します。
 - d) **OK** をクリックして、**操作の選択** ダイアログ ボックスを閉じます。Silk Test Workbench は、記録した操作にその操作を追加し、モバイル デバイスまたはエミュレータ上でそれを再生します。
- 記録中、Silk Test Workbench は記録ウィンドウの隣にマウスの位置を表示します。その表示をクリックすると、デバイス画面に絶対的な位置とアクティブ オブジェクトに相対的な位置を切り替えることができます。記録中に利用可能なアクションについての詳細は、「記録中に利用可能なアクション」を参照してください。
- 10 **停止** をクリックします。**記録完了** ダイアログ ボックスが開きます。
 - 11 **OK** をクリックします。

テストが期待通りの動作をするか確認するためにテストを再生します。必要な場合には変更をするために、テストを編集することも可能です。

UI オートメーション メソッドの動的呼び出し

UI オートメーション コントロールのサポートするメソッドが、すべてのサポートするテクノロジーの対応するコントロールに対して有効であることを保障するため、Silk Test Workbench API がサポートするのは、これらのコントロールに対して利用できるメソッドやプロパティのサブセットになります。コントロールに対して Silk Test Workbench API で利用できないその他のメソッドやプロパティの呼び出しには、動的呼び出しを使用してください。

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある setTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型。

Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point、Rect など) が含まれます。

- 列挙型。

列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが .NET 列挙型 System.Windows.Visibility のパラメータを必要とする場合には、Visible、Hidden、Collapsed の文字列値を使用できます。

- .NET 構造体とオブジェクト。

.NET 構造体とオブジェクトパラメータはリストとして渡します。リスト内の要素は、テスト アプリケーションの .NET オブジェクトで定義されているコンストラクタの 1 つと一致しなければなりません。たとえば、メソッドが .NET 型 System.Windows.Vector のパラメータを必要とする場合、2 つの整数値を持つリストを渡すことができます。これが機能するのは、System.Windows.Vector 型が 2 つの整数値を引数に取るコンストラクタを持つためです。

- その他のコントロール。

コントロールパラメータは TestObject として渡すことができます。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- MSDN が定義する AutomationElement クラスのすべてのパブリック メソッドとプロパティ。詳細については、<http://msdn.microsoft.com/en-us/library/system.windows.automation.automationelement.aspx> を参照してください。
- MSUIA が公開するすべてのメソッドとプロパティ。利用可能なメソッドとプロパティは「パターン」で分類されます。パターンとは、MSUIA 固有の用語です。すべてのコントロールは、いくつかのパターンを実装します。一般的なパターンについての概要およびすべての利用可能なパターンについては、<http://msdn.microsoft.com/en-us/library/ms752362.aspx> を参照してください。カスタムコントロールの開発者は、MSUIA パターン セットを実装することによって、カスタムコントロールのテストサポートを提供できます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。
- すべてのその他の型の場合は文字列。

この文字列表現を取得するには、テスト対象アプリケーションの返された .NET オブジェクトに対して ToString メソッドを呼び出します。

例

この例では、動的呼び出しを使って UIADocument コントロールのスクロール メソッドを呼び出す方法を紹介합니다。Silk Test Workbench の API ではこのようなスクロール メソッドを提供していません。これは、これらのメソッドが、UI オートメーショ

プロバイダー インターフェイスを実装するすべてのテクノロジーの UIADocument コントロールで利用可能ではないためです。

コントロールに対して利用できるメソッドとプロパティを表示するには、以下に示すようなコードを使用できます。

```
' VB .NET code
Dim textBox As UIADocument = mainWindow.Find("//UIADocument")
Dim propertyList = textBox.GetPropertyList()
Dim methodList = textBox.GetDynamicMethodList()
```

この例では、GetPropertyList メソッドが返す *propertyList* には、ScrollPattern.VerticalScrollPercent プロパティが含まれます。GetDynamicMethodList メソッドが返す *methodList* には、ScrollPattern.ScrollVertical メソッドが含まれます。

動的呼び出しを使うと、次のようなコードで ScrollPattern.ScrollVertical メソッドを呼び出すことができます。

```
textBox.Invoke("ScrollPattern.ScrollVertical", ScrollAmount.SmallIncrement)
```

または、次のようなコードで ScrollPattern.VerticalScrollPercent プロパティを呼び出すことができます。

```
textBox.GetProperty("ScrollPattern.VerticalScrollPercent")
```

UI オートメーションを使ってコントロールを識別するためのロケータ属性

UI オートメーション プロバイダー インターフェイスを実装した Windows ベースのアプリケーションのコントロールに対して、次のロケータ属性がサポートされています。

- *automationId*
- *caption*
- *className*
- *name*
- すべての動的ロケータ属性



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

UI オートメーション プロバイダー インターフェイスを実装した Windows ベースのアプリケーションのコントロールを識別するには、スクリプトで *automationId*、*caption*、*className*、*name*、またはすべての動的ロケータ属性を使用できます。*automationId* はアプリケーション開発者が設定します。たとえば、*automationId* を持つロケータは、以下ようになります: //

```
UIAButton[@automationId="okButton"]
```

automationId は一般に非常に有用で安定した属性であるため、使用することを推奨します。

属性の種類	説明	例
<i>automationId</i>	テスト対象アプリケーションの開発者によって設定される識別子。Visual Studio デザイナは、デザイナー上で作成されたすべてのコントロールに自動的に <i>automationId</i> を割り当てます。アプリケーション開	// UIAButton[@automationId="okButton"]

属性の種類	説明	例
caption	発者は、アプリケーションのコード上でコントロールを識別するために、この ID を使用します。 コントロールが表示するテキスト。複数の言語にローカライズされたアプリケーションをテストする場合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	//UIButton[@caption="Ok"]
className	UI オートメーション コントロールのクラス名 (名前空間なし)。 <i>className</i> 属性を使用すると、Silk Test Workbench が解決する標準 UI オートメーション コントロールから派生したカスタム コントロールを識別するのに役立ちます。	// UIButton[@className='MyCustomButton']
name	コントロールの名前。テスト対象アプリケーションの開発者によって設定されます。	//UIButton[@name="okButton"]

Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性を、この表に示した順番に使用して UI オートメーション コントロールのロケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、*automationId* を使って固有に識別できれば、Silk Test Workbench によるロケータの作成に使用されます。

ユーザーの AUT の UI オートメーション コントロールに対して使用できるその他のカスタム属性を確認するには、**プロパティの検証** ダイアログ ボックスを使用できます。使用するときには、記録する UI オートメーション コントロール上にマウス カーソルを移動し、**Ctrl + Alt** をクリックします。これで、コントロールに対して使用できるプロパティを確認できます。たとえば、アプリケーションによっては、属性 value が有効な場合もあります。

UI オートメーション コントロールでのスクロール

Silk Test Workbench では、UI オートメーション コントロールに応じて、2 種類のスクロール方法とプロパティを提供します。

- 1 つめの種類のコントロールには、それ自体でスクロール可能なコントロールが含まれ、スクロールバーは子として明示的に表示されません。たとえば、コンボ ボックス、ペイン、リスト ボックス、ツリー コントロール、データ グリッド、オート コンプリート ボックスなどがあります。
- 2 つめの種類のコントロールには、それ自体ではスクロール不可能なコントロールが含まれ、スクロール用にスクロールバーが子として表示されます。たとえば、テキスト フィールドがあります。

Silk Test Workbench にこのような違いがあるのは、UI オートメーション コントロールがこの 2 通りの方法でスクロールを実装するためです。

スクロールをサポートするコントロール

この場合、スクロール方法とプロパティは、スクロールバーを含むコントロールで使用できます。したがって、Silk Test Workbench ではスクロールバー オブジェクトは表示されません。

例

以下のコマンドでは、リスト ボックスが一番下までスクロールされます。

```
listBox.SetVerticalScrollPercent(100)
```

以下のコマンドでは、リスト ボックスが 1 ユニットずつ下方へスクロールされます。

```
listBox.ScrollVertical(ScrollAmount.SmallIncrement)
```

スクロールをサポートしないコントロール

この場合、スクロールバーが表示されます。コントロール自体で可能なスクロール方法とプロパティはありません。水平スクロールバーと垂直スクロールバーの各オブジェクトを使用すると、対応する API 関数

でパラメータとして増分または減分、または最終位置を指定することでコントロール内をスクロールできます。増分または減分として ScrollAmount 列挙の値を使用できます。詳細については、MSUIA のドキュメントを参照してください。最終位置は、オブジェクトの位置に関連し、アプリケーション設計者によって定義されます。

例

以下のコマンドでは、テキスト ボックス内の垂直スクロールバーが 15 の位置までスクロールされます。

```
textBox.UIAVerticalScrollBar().ScrollToPosition(15)
```

以下のコマンドでは、テキスト ボックス内の垂直スクロールバーが一番下までスクロールされます。

```
textBox.UIAVerticalScrollBar().ScrollToMaximum()
```

UI オートメーション使用時の制限事項

UI オートメーション使用時における既知の制限事項を以下に示します。

UI オートメーション サポート使用時に IME が正しく処理されない

UI オートメーション サポートを有効にすると、Silk Test Workbench の記録/再生時に IME を伴うキーの入力が正しく処理されません。

UI オートメーション サポートを使ったテスト時のトラブルシューティング

Microsoft Windows 7 で記録した UI オートメーション コントロールを含むスクリプトを Microsoft Windows 8 以降で再生できない理由

Microsoft Windows 7 以前で UI オートメーション コントロールを含んだスクリプトを記録すると、Microsoft Windows 8 以降での再生に失敗する場合があります。これは、Microsoft がオートメーションの仕組みを変更し、UI オートメーションの動作が Windows のバージョン間で異なることが原因です。

たとえば、アプリケーションのある UI オートメーション コントロールでは、Microsoft Windows 7 では automationId 属性に値が設定されていますが、Microsoft Windows 10 では値が設定されていません。

このような場合、Microsoft Windows の新しいバージョンでスクリプトを記録しなおすことを Micro Focus では推奨しています。

Microsoft Office アプリケーションの最初の操作が再生されない理由

Microsoft Office アプリケーションの記録時に、Silk Test Workbench で記録中の操作が正しくアプリケーションで再生されるためには、アプリケーション ウィンドウがアクティブである必要があります。たとえば、Microsoft Excel の記録時に、他のアプリケーションにフォーカスを移した後で、Excel のセルに対するクリックを Silk Test Workbench で記録すると、Excel 上ではセルのクリックが再生されません。このような状況になった場合は、記録した操作を繰り返した後で、余計な操作をテストから削除する必要があります。

この問題を回避するには、操作を記録する前に Microsoft Office アプリケーションのタイトル バーを左クリックして、アプリケーションをアクティブにします。タイトル バーのクリックは Silk Test Workbench は記録しないため、そのまま記録を続けられます。

Silk Test Workbench での Unicode コンテンツのサポートの概要

Open Agent は、Unicode 対応済みです。つまり、Open Agent は、2 バイト (ワイド) 言語を認識できません。

Silk Test Workbench を使用して、中国語、韓国語、日本語 (漢字) などの 2 バイト言語や、それらを組み合わせたコンテンツを含んだアプリケーションをテストできます。

Open Agent は、以下をサポートします。

- Windows のローカライズ版。
- 国際化キーボードとネイティブ言語の入力方式エディター (IME)。
- テストケース、メソッドなどにパラメータとして国際化文字列を渡す、および文字列の比較。
- 複数の形式でのテキスト ファイルの読み書き : ANSI、Unicode、UTF-8。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

Silk Test Workbench を使用した 2 バイト文字のテストを行う前に

国際化されたアプリケーション、特に 2 バイト文字を含んだアプリケーションをテストするのは、英語 (1 バイト文字) のみを含んだアプリケーションをテストするよりも複雑です。国際化アプリケーションのテストにおいては、オペレーティング システムのサポートから、言語パック、フォント、IME の動作、さらに複合した言語など、さまざまな問題を理解する必要があります。

Silk Test Workbench を使用してアプリケーションのテストを始める前に、以下を確認する必要があります。

- 必要なローカライズ OS、地域の設定、必要な言語パックがテスト対象アプリケーション (AUT) の要求を満たしているか。
- AUT を表示するのに必要なフォントがインストールされているか。
- データ入力に IME が必要なアプリケーションをテストする場合、適切な IME がインストールされているか。

テキスト解決のサポート

テキスト解決メソッドとイメージ解決メソッドを使用して、オブジェクト解決で識別できない、高度にカスタマイズされたコントロールを含むテスト アプリケーションを便利に操作できます。座標ベースのクリックの代わりにテキストクリックを使用し、コントロール内に指定されたテキスト文字列をクリックできます。

たとえば、次の表の 2 行目の最初のセルを選択することをシミュレートできます。


CustomerName	FirstOrder	ID	IsActive	CreditCard
Bob Villa	01.01.2008	0	<input checked="" type="checkbox"/>	MasterCard
Brian Miller	02.01.2008	1	<input type="checkbox"/>	Visa
Caral Rudd	03.01.2008	2	<input checked="" type="checkbox"/>	American Ex...
Dan Rundgren	04.01.2008	3	<input type="checkbox"/>	MasterCard
Devie Yingstein	05.01.2008	4	<input checked="" type="checkbox"/>	Visa

セルのテキストを指定すると、次のコードが生成されます :

```
'VB code  
table.TextClick("Brian Miller")
```

テキスト解決メソッドは、次のテクノロジー ドメインでサポートされます。

- Win32
- WPF
- Windows Forms
- Java SWT と Eclipse
- Java AWT/Swing

 **注:** Java アプレット、および Java バージョンが 1.6.10 より前の Swing アプリケーションの場合、テキスト解決は追加設定なしでサポートされます。Direct3D をサポートしない Java バージョン 1.6.10 以降の Swing アプリケーションの場合は、アプリケーションの起動時に次のコマンドライン要素を追加する必要があります。


```
-Dsun.java2d.d3d=false
```


例 :

```
javaw.exe -Dsun.java2d.d3d=false -jar mySwingApplication.jar
```

Direct3D をサポートする Java アプレットや Swing アプリケーションでは、テキスト解決はサポートされません。


- Internet Explorer
- WebDriver ベースのブラウザ


 **注:** テキスト解決は、画面に表示されていないコントロールに対しては機能しません。たとえば、スクロールして画面外にあるテキストに対しては、テキスト解決を使用できません。

 **注:** 対象のテキストで使用されたフォントがテストが実行されるマシン上にインストールされていない場合、テキスト解決が機能しない場合があります。

WebDriver ベースのブラウザ

テキスト解決メソッドは、BrowserWindow オブジェクトと DomElement オブジェクトに対して実行できます。

 **注:** <canvas> 要素に描画されたテキストに対しては、テキスト解決は機能しません。

 **注:** ::before や ::after などの CSS 擬似要素によって追加されたコンテンツに対しては、テキスト解決は機能しません。

テキスト解決メソッド

Silk Test Workbench では、AUT がスクリーン上に描画するテキストを処理してテストを実行できるように、次のメソッドを提供しています。

TextCapture コントロール内のテキストを返します。子コントロールのテキストも返します。

TextClick コントロール内の指定テキストをクリックします。テキストが検出されるか、同期オブションで定義できるオブジェクト解決タイムアウトに達するまで待機します。

TextRectangle コントロール内の特定テキストの矩形、またはコントロールの領域を返します。

TextExists コントロール内またはコントロールの領域内に特定テキストが存在するかどうかを判断します。

テキスト解決メソッドでは、部分的に一致する単語よりも完全に一致する単語が優先されます。Silk Test Workbench では、完全に一致する単語の前に部分的に一致する単語が画面に表示されていても、部分的に一致する単語よりも完全に一致した単語の出現が先に解決されます。完全に一致する単語がない場合は、部分的に一致する単語が画面に表示される順序で使用されます。

TextClick、TextRectangle、TextExists メソッドは、その内部で TextCapture を使用してアプリケーションから表示されたテキストを取得し、テキストを処理します。TextCapture メソッドは、2 種類の方法で実装されています。Silk Test Workbench は、テスト対象アプリケーションの種類に応じて実装を使い分けます。

- ネイティブ Windows アプリケーション (WPF、WinForms、Java アプリケーション)、および Internet Explorer の場合は、Silk Test Workbench は Windows API のテキスト描画関数をフックして、アプリケーションが画面に描画するテキストを取得します。
- Google Chrome、Mozilla Firefox、Microsoft Edge、Apple Safari の場合は、Silk Test Workbench は JavaScript ベースの方法を使って、ブラウザが描画したテキストを取得します。



注: これら 2 種類の実装方法の違いにより、同じ Web アプリケーションでも、使用するブラウザに応じて異なるテキストを Silk Test Workbench が返す場合があります。

例

ユーザー インターフェイスには、テキスト「*the hostname is the name of the host*」が表示されているとします。次のコードでは、画面上では *host* の前に *hostname* が表示されていますが、*hostname* ではなく *host* がクリックされます。

```
'VB code
control.TextClick("host")
```

次のコードでは、2 番目の出現箇所であることを指定することで、単語 *hostname* 中の部分文字列 *host* がクリックされます。

```
'VB code
control.TextClick("host", 2)
```

カスタム コントロール

Silk Test Workbench では、カスタム コントロールを扱うときに、以下の機能がサポートされます。

- 動的呼び出しを使用すると、テスト対象アプリケーション (AUT) 内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を Silk Test Workbench で直接実行できます。
- クラス マッピング 機能によって、カスタム コントロール クラスの名前を標準 Silk Test クラスの名前にマップできます。このようにすると、標準 Silk Test クラスでサポートされる機能をテストで使用できます。

Silk Test Workbench は、次のテクノロジ ドメインに対する UI のカスタム コントロールの管理をサポートします。

- Win32
- Windows Presentation Foundation (WPF)
- Windows Forms
- Java AWT/Swing
- Java SWT
- AUT にコードを追加して、カスタム コントロールをテストできます。
- **カスタム コントロールの管理** ダイアログ ボックスを使用して、ロケータで利用できるカスタム コントロールの名前を指定したり、カスタム コントロールを操作する再利用可能なコードを作成することができます。




注: カスタム コントロールでは、Click、TextClick、TypeKeys などのメソッドだけが、Silk Test Workbench で記録できます。Apache Flex アプリケーションをテストする場合を除き、カスタム コントロールのカスタム メソッドは記録できません。

動的呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すこ

とができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある SetTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

 **注:** DOM 要素のメソッドを動的に呼び出すことはできません。

動的呼び出しに関するよくある質問

このセクションでは、カスタム コントロールをテストするために動的にメソッドを呼び出すときの質問を示します。

Invoke メソッドを使用して呼び出せるメソッド

特定のテスト オブジェクトに対して、Invoke メソッドを使用して呼び出せるすべてのメソッドのリストを取得するには、GetDynamicMethodList を使用します。リストを表示するには、コンソールに出力したり、デバッガーで表示することなどができます。

呼び出しで複雑なオブジェクトが返されることが期待されるときに単純な文字列が返される理由

Invoke メソッドは単純なデータ型のみを返すことができます。複雑な型は文字列として返されます。Silk Test Workbench は ToString メソッドを使用して、戻り値の文字列表現を取得します。個々のメソッドを呼び出し、最初のメソッドの呼び出しで返される複雑なオブジェクトのプロパティを読み取るには、Invoke ではなく、InvokeMethods を使用します。

複数の InvokeMethods 呼び出しを使用するときにスクリプトを単純化する方法

スクリプトで大量の InvokeMethods を使用すると、すべてのメソッド名を文字列として渡し、すべてのパラメータをリストとして渡す必要があるため、複雑になります。このような複雑なスクリプトを単純化するには、InvokeMethods を通じてコントロールを操作するのではなく、AUT の実際のコントロールを操作する静的メソッドを作成します。詳細については、「テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする」を参照してください。

テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする

Windows Forms アプリケーションまたは WPF アプリケーションをテストし、複雑なカスタム コントロールまたは Invoke および InvokeMethods メソッドを使用するだけではテストできないカスタム コント

ロールをテストする場合は、テスト対象アプリケーション (AUT) の実際のコントロールを操作する静的メソッドを作成し、このコードを AUT に追加できます。

AUT にコードを追加することのメリットは、AUT のコードで、動的呼び出しメソッドによるメソッド呼び出しのリフレクション形式ではなく、通常のメソッド呼び出しを使用してコントロールを操作できるという点です。そのため、コードを作成する時に、コード補完と IntelliSense を使用できます。その後、AUT のコードを単純な呼び出しで呼び出し、該当するコントロールをパラメータとして渡すことができます。

AUT にコードを追加するには、次の方法があります。

- AUT でコードをコンパイルします。実装は簡単ですが、意図しない AUT の変更を行うことになりま
- テスト スクリプトの LoadAssembly メソッドを使用して、実行時にコードを AUT に挿入します。AUT でコードをコンパイルする場合よりも作業は多くなりますが、挿入されたコードはテストコードの近くに配置されます。LoadAssembly は、WPFWindow クラスおよび FormsWindow クラスで使用できます。

例 : UltraGrid Infragistics コントロールのテスト

この例では、UltraGrid コントロールの内容を取得する方法を示します。UltraGrid コントロールは、Infragistics が提供する NETAdvantage for Windows Forms ライブラリに含まれています。ライブラリの試用版を <http://www.infragistics.com/products/windows-forms/downloads> からダウンロードできます。

UltraGridUtil クラスを作成するには、以下の操作を実行します。

1. 新しいテスト スクリプトを作成します。新しいスクリプト AUTExtensions を呼び出します。



注: AUT は、.NET バージョン 4 以降を使用する必要があります。.NET スクリプトをコンパイルすると .NET 4.0 アセンブリが作成されますが、このアセンブリは使用している .NET バージョンが同じかより新しい AUT でのみロードできます。AUT でバージョン 4.0 以前の .NET を使用している場合は、Visual Studio を使用して、UltraGridUtil クラスを含むクラス ライブラリ プロジェクトを作成してください。

2. 必要な依存関係への参照をスクリプトに追加します。たとえば、Infragistics バージョン 12.2 の場合、次のアセンブリへの参照が必要です。

- Infragistics4.Shared.v12.2
- Infragistics4.Win.UltraWinGrid.v12.2
- Infragistics4.Win.v12.2

AUT で使用している Infragistics のバージョンが不明な場合は、Microsoft の **Process Explorer** ツールを使用して、AUT にロードされているアセンブリを確認できます。

- a. AUTExtensions スクリプトの内容を次のコードに置き換えます。

```
' VB code
Public Class UltraGridUtil

    Public Shared Function GetContents(ultraGrid As
Infragistics.Win.UltraWinGrid.UltraGrid) As List(Of List(Of String))
        Dim contents = New List(Of List(Of String))
        For Each row In ultraGrid.Rows
            Dim rowContents = New List(Of String)
            For Each cell In row.Cells
                rowContents.Add(cell.Text)
            Next
```

```
        contents.Add(rowContents)
        Next
        Return contents
    End Function

End Class
```



注: Shared 修飾子によって、GetContents メソッドが静的メソッドになります。

3. 再生中に、AUT にアセンブリをロードします。

- 既存のテスト スクリプトを開くか、新しいテスト スクリプトを作成します。
- AUTExtensions スクリプトへの参照を追加します。
- 次のコードをテスト スクリプトに追加します：

```
' VB code
mainWindow.LoadAssembly(GetType(UltraGridUtil).Assembly.Location)
```



注: Silk Test Workbench によって、各スクリプトが個別の .NET アセンブリにコンパイルされます。

4. 挿入したコードの静的メソッドを呼び出して、UltraGrid の内容を取得します：

```
'VB code
Dim ultraGrid = mainWindow.Control("@automationId='my grid'")
Dim contents As IList =
mainWindow.Invoke("UltraGridUtil.GetContents", ultraGrid)
```

AUT へのコードの追加に関するよくある質問

このセクションでは、カスタム コントロールをテストするために AUT にコードを追加するときの質問を示します。

LoadAssembly メソッドを使用して AUT に挿入したコードが AUT で更新されない理由

AUT 内のコードが、LoadAssembly メソッドを使用して AUT に挿入したコードによって置き換えられない場合、アセンブリがすでに AUT にロードされている可能性があります。アセンブリをアンロードすることはできないため、AUT を閉じてから、再開する必要があります。

メソッドを呼び出すと入力引数の型が一致しない理由

何らかのメソッドを呼び出したときに、入力引数の型が一致しないことを示すエラーが表示される場合は、呼び出すメソッドは見つかりましたが、引数が正しくありません。スクリプトで正しいデータ型を使用していることを確認します。

スクリプトで LoadAssembly メソッドを使用してアセンブリを AUT にロードする場合にこのエラーが発生するもう 1 つの理由として、AUT が使用するバージョンとは異なるサードパーティ ライブラリのバージョンに対してアセンブリが作成されている可能性があります。この問題を修正するには、スクリプトで参照されているアセンブリを変更します。AUT で使用されているサードパーティ ライブラリのバージョンが不明な場合は、Microsoft の **Process Explorer** ツールを使用できます。

アセンブリをコピーできないときにコンパイル エラーを修正する方法

LoadAssembly メソッドで AUT にコードを追加しようとしたときに、次のコンパイル エラーが発生することがあります。

```
Could not copy '<assembly_name>.dll' to '<assembly_name>.dll'. The process cannot access the file.
```

このコンパイル エラーは、アセンブリがすでに AUT にロードされていて、上書きできないために発生します。

このコンパイルエラーを修正するには、AUT を閉じて、再度スクリプトをコンパイルします。最初から問題が発生しないようにするには、一時ディレクトリなどを使用してアセンブリをスクリプト内にコピーし、コピーしたアセンブリを LoadAssembly メソッドで挿入します。例：

```
' VB code
Dim typeThatContainsYourStaticMethod = GetType (UltraGridUtil)
Dim assemblyToLoadIntoAUT = typeThatContainsYourStaticMethod.Assembly.Location
Dim tempFile = IO.Path.GetTempFileName()
IO.File.Copy(assemblyToLoadIntoAUT, tempFile, True )
mainWindow.LoadAssembly(tempFile)
```

Apache Flex カスタム コントロールのテスト

Silk Test Workbench では、Apache Flex カスタム コントロールのテストがサポートされています。ただし、デフォルトでは、Silk Test Workbench は、カスタム コントロールの個別のサブコントロールを記録および再生することはできません。

カスタム コントロールをテストする場合、以下のオプションが存在します。

- 基本サポート

基本サポートでは、動的呼び出しを使用して、再生中にカスタム コントロールと対話します。作業量が少なく済むこのアプローチは、テスト アプリケーションにおいて、Silk Test Workbench が公開しないカスタム コントロールのプロパティおよびメソッドにアクセスする場合に使用します。カスタム コントロールの開発者は、コントロールのテストを容易にすることのみを目的としたメソッドおよびプロパティをカスタム コントロールに追加することもできます。ユーザーは、動的呼び出し機能を使用してこれらのメソッドやプロパティを呼び出すことができます。

基本サポートには以下のような利点があります。

- 動的呼び出しでは、テスト アプリケーションのコードを変更する必要がありません。
- 動的呼び出しを使用することによって、ほとんどのテストのニーズを満たすことができます。

基本サポートには以下のような短所があります。

- ロケーターには、具体的なクラス名が組み込まれません (たとえば、Silk Test Workbench では「// FlexSpinner」ではなく「//FlexBox」と記録されます)。
- 記録のサポートが限定されます。
- Silk Test Workbench では、イベントを再生できません。

例を含む動的呼び出しの詳細については、「*Apache Flex* メソッドの動的呼び出し」を参照してください。

- 高度なサポート

高度なサポートでは、カスタム コントロールに対して、特定のオートメーション サポートを作成できます。この追加のオートメーション サポートによって、記録のサポートおよびより強力な再生のサポートが提供されます。高度なサポートには以下のような利点があります。

- イベントの記録と再生を含む、高レベルの記録および再生のサポートが提供されます。
- Silk Test Workbench では、カスタム コントロールが他のすべての組み込み Apache Flex コントロールと同様に処理されます。
- Silk Test Workbench API とシームレスに統合できます。
- Silk Test Workbench では、ロケーターで具体的なクラス名が使用されます (たとえば、Silk Test Workbench では「//FlexSpinner」と記録されます)。

高度なサポートには以下のような短所があります。

- 実装作業が必要です。テスト アプリケーションを変更し、Open Agent を拡張する必要があります。

カスタム コントロールの管理

Silk Test Workbench が専用サポートを提供していないカスタム コントロールに対応するカスタム クラスを作成できます。カスタム クラスを作成すると、以下の利点があります。

- スクリプトおよびビジュアル テストのロケーターが効率化されます。
- カスタム コントロールと対話するための再利用可能コードを簡単に記述できます。

例 : UltraGrid Infragistics コントロールのテスト

カスタム グリッド コントロールが Silk Test Workbench で汎用クラス Control として認識されるとします。Silk Test Workbench のカスタム コントロール サポートを使用すると、以下の利点があります。

カスタム コントロール クラス名をロケーターで使用できるため、オブジェクトの認識率が高まります。

複数のオブジェクトが Control として認識されることがあります。ロケーターには、特定のオブジェクトを識別するためのインデックスが必要です。たとえば、オブジェクトはロケーター // Control[13] を使用して識別できます。このコントロールのカスタム クラス (クラス UltraGrid など) を作成する場合は、ロケーター //UltraGrid を使用できます。カスタム クラスを作成することによって、テスト対象アプリケーションが変更された場合にオブジェクト識別子が変わりやすい、大きな数字のインデックスを使用する必要がなくなります。

スクリプト内のコントロールに、再利用可能な再生操作を実行できます。

カスタム クラスを使用している場合、ユーザー インターフェイスにカスタム コントロールを指定すると生成されるクラスであるカスタム クラスに以下のコードを追加することで、グリッドのコンテンツをメソッド内に取り込む動作をカプセル化できます。

通常は、以下のいずれかの方法で、メソッドをカスタム コントロール クラスに実装できます。

- Click、TypeKeys、TextClick、および TextCapture などのメソッドを使用できます。
- AUT のオブジェクトで動的にメソッドを呼び出せます。
- AUT に追加したメソッドを動的に呼び出せます。これは、この例で説明されている手法です。

以下のコードを使用して、「テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする」の例で定義されている静的メソッドを呼び出すことができます。メソッド GetContents が、生成されたクラス UltraGrid に追加されます。

```
' VB code
Partial Public Class UltraGrid

    Public Function GetContents() As IList
        Return
```

```
Invoke("UltraGridUtil.GetContents", Me)
End Function
```

```
End Class
```

クラスをカスタム コントロールとして定義すると、Dialog クラスのように、すべての組み込みクラスの場合と同じ方法でそのクラスを使用できません。

```
' VB code
Dim ultraGrid As UltraGrid =
mainWindow.UltraGrid("@automationId='my
grid'")
Dim contents = ultraGrid.GetContents()
```

カスタム コントロールのサポート


Silk Test Workbench は、次のテクノロジー ドメインに対する UI のカスタム コントロールの管理をサポートします。

- Win32
- Windows Presentation Foundation (WPF)
- Windows Forms
- Java AWT/Swing
- Java SWT

Silk Test Workbench が専用サポートを提供していないカスタム コントロールに対応するカスタム クラスを作成するには、以下を実行します。


1. ツール > **カスタム コントロールの管理** をクリックします。**カスタム コントロールの管理** ダイアログボックスが開きます。
2. **カスタム コントロール用 .NET スクリプト** フィールドで、任意の名前を入力するか、**参照** をクリックして、**カスタム コントロールを含めるスクリプト** を選択します。

カスタム コントロール用の .NET スクリプトは、Common プロジェクトに保存されます。すべての新しいデータベースに対して、デフォルトの .NET スクリプトは *CustomControlScript* という名前です。このスクリプトは、ユーザーが最初にデータベースにログインしたときに自動的に作成されます。

 **注:** .NET スクリプトは必須です。**カスタム コントロール用 .NET スクリプト** フィールドでスクリプトが選択されていない場合、**OK** ボタンは使用できません。ボタンを有効にするには、スクリプトを選択します。既存のデータベースの場合、Common プロジェクトの書き込み権があり、カスタム コントロール セット用の .NET スクリプトがない場合には、Silk Test Workbench は Common プロジェクトに新しいスクリプトを保存しようとします。*CustomControlScript* という名前のスクリプトが既に存在し、Common プロジェクトの書き込み権がある場合、Silk Test Workbench はスクリプトの名前の終わりに番号を追加してスクリプトを保存しようとします (*CustomControlScript2*、*CustomControlScript3* など)。

3. 新しいカスタム クラスを作成するテクノロジー ドメインのタブをクリックします。
4. コントロールを自動的に選択する場合は、**識別** をクリックして Silk Test Workbench を識別モードに切り替えると、通常通りにコントロールを選択できるようになります。**カスタム コントロールのクラス名** と **Silk Test 基本クラス** フィールドは、特定されたコントロールに基づいて、現在のタブに入力されます。それ以外の場合、以下の **追加** ステップを続けます。
5. **追加** をクリックします。新しい行がカスタム コントロールのリストに追加されます。
6. カスタム コントロールを手動でリストに追加するように選択した場合は、以下を実行します。
 - a) **Silk Test 基本クラス** 列で、クラスの取得元となる既存の基本クラスを選択します。
このクラスは、ご使用のカスタム コントロールのタイプに最も一致率が高くなければなりません。

- b) **カスタム コントロール クラス名** 列に、マップしているクラスの完全修飾クラス名を入力します。
たとえば、Infragistics.Win.UltraWinGrid.UltraGrid です。クラス名には、ワイルドカード ? および * を使用できます。
- c) **Win32** で作業していて、クラスの参照方法を変更する場合は、次の列のチェック ボックスをオンにします。チェック ボックスをオフにした場合、名前はデフォルトで **Silk Test 基本クラス** と同じになります。 **Silk Test クラス** 列が有効になります。
- d) **Silk Test クラス** 列で、**F2** をクリックし、クラスの参照に使用する名前を入力します。
この名前は、ロケーターに表示されます。たとえば、//Control[13] でなく //UltraGrid を入力します。

 **注:** 有効なクラスを追加すると、そのクラスは **Silk Test 基本クラス** リストで使用できるようになります。追加したクラスは、基本クラスとして再使用できます。


7. Win32 アプリケーションの場合のみ：**クラスの宣言を使用する** 列で、値を **False** に設定して、カスタム コントロール クラスの名前を標準 Silk Test クラスの名前に単純にマップします。


カスタム コントロール クラスを標準 Silk Test クラスにマップすると、テストの際に標準 Silk Test クラスでサポートされている機能を使用できます。カスタム コントロール クラスのクラス宣言を追加して使用する場合は、この値を **True** にします。

8. **OK** をクリックします。

9. スクリプトの場合のみ：

- a) カスタム コントロール用のクラスにカスタム メソッドおよびプロパティを追加します。
b) スクリプト内で新しいクラスのカスタム メソッドおよびプロパティを使用します。

 **注:** カスタム メソッドおよびプロパティは記録されません。

 **注:** スクリプト ファイル内のカスタム クラスまたは基本クラスの名前を変更しないでください。スクリプト内に生成されたクラスを変更した場合、予期しない動作を起こすことがあります。カスタム クラスにプロパティおよびメソッドを追加する場合にのみスクリプトを使用してください。それ以外の変更をカスタム クラスに加える場合は **カスタム コントロールの管理** ダイアログ ボックスを使用してください。

カスタム コントロール オプション


ツール > カスタム コントロールの管理。

Silk Test Workbench は、次のテクノロジー ドメインに対する UI のカスタム コントロールの管理をサポートします。

- Win32
- Windows Presentation Foundation (WPF)
- Windows Forms
- Java AWT/Swing
- Java SWT

カスタム コントロール用 .NET スクリプト で、新しいカスタム クラスをその中に生成するスクリプト ファイルを定義します。

カスタム コントロール用の .NET スクリプトは、Common プロジェクトに保存されます。すべての新しいデータベースに対して、デフォルトの .NET スクリプトは *CustomControlScript* という名前です。このスクリプトは、ユーザーが最初にデータベースにログインしたときに自動的に作成されます。

 **注:** .NET スクリプトは必須です。 **カスタム コントロール用 .NET スクリプト** フィールドでスクリプトが選択されていない場合、**OK** ボタンは使用できません。ボタンを有効にするには、スクリプトを選択します。既存のデータベースの場合、Common プロジェクトの書き込み権があり、カスタム コントロール セット用の .NET スクリプトがない場合には、Silk Test Workbench は Common プロジェクトに新しいスクリプトを保存しようとします。 *CustomControlScript* という名前のスクリプトが既に存在し、Common プロジェクトの書き込み権がある場合、Silk Test Workbench はスクリプ

トの名前の終わりに番号を追加してスクリプトを保存しようとして (CustomControlScript2、CustomControlScript3 など)。

カスタム コントロール クラスを標準 Silk Test クラスにマップすると、テストの際に標準 Silk Test クラスでサポートされている機能を使用できます。次の **カスタム コントロール** オプションが使用できます。

オプション


説明

Silk Test 基本クラス 自分のクラスの派生元として使用する既存の基本クラスを選択します。このクラスは、ご使用のカスタム コントロールのタイプに最も一致率が高くなければなりません。

Silk Test クラス クラスの参照に使用する名前を入力します。この名前は、ロケーターに表示されます。

カスタム コントロールのクラス名 マッピングされているクラスの完全修飾クラス名を入力します。クラス名には、ワイルドカード ? および * を使用できます。

クラスの宣言を使用する このオプションは Win32 アプリケーションの場合のみ使用できます。デフォルト値は False で、カスタム コントロール クラスの名前が標準 Silk Test クラスの名前にマップされることを意味します。カスタム コントロール クラスのクラス宣言を追加して使用する場合は、この設定を True にします。

 **注:** 有効なクラスを追加すると、そのクラスは **Silk Test 基本クラス** リストで使用できるようになります。追加したクラスは、基本クラスとして再使用できます。

例 : UltraGrid Infragistics コントロールのオプションの設定

UltraGrid Infragistics コントロールをサポートするには、次の値を使用します。

オプション	値
Silk Test 基本クラス	Control
Silk Test クラス	UltraGrid
カスタム コントロールのクラス名	Infragistics.Win.UltraWinGrid. UltraGrid


実行時間の計測

Timer クラスが提供するメソッドやプロパティを使用して、テストの実行にかかる時間を計測できます。詳細については、「[Timer クラス](#)」を参照してください。

特に、これらのメソッドとプロパティは、Silk Performer から呼び出されるテスト実行の計測に使用できるという利点があります。Silk Test Workbench と Silk Performer の統合についての詳細は、『[Silk Performer ヘルプ](#)』を参照してください。

テスト実行の遅延

テスト対象アプリケーションによっては、UI でのアプリケーション データの読み込みに多くの時間を必要とするため、テストの再生に必要なオブジェクトのロードが時間内に終わらない場合があります。このような AUT でテストの再生を正しく行うには、操作を実行する前にオブジェクトの存在を確認したり、操作の実行前にスリープを挿入する必要があります。

 **注:** Micro Focus では、テストにスリープを追加することは基本的には推奨していません。たいていの場合、オブジェクトが利用可能かどうかを Silk Test Workbench が自動的に検出するので、スリープはテストのパフォーマンスを落とす結果になるためです。

1. オブジェクトが AUT で利用可能かどうかを確認するには、Exists メソッドを使用します。
たとえば、INPUT ボタンが利用可能になるまで 6 秒間待機する場合は、テスト スクリプトに次の行を追加します。

```
'VB .NET code  
browserWindow.Exists("//INPUT", 6000)
```


2. コントロールの操作を実行する前にスリープを追加するには、Sleep メソッドを使用します。
たとえば、6 秒間スリープする場合は、テスト スクリプトに次の行を追加します。


```
'VB .NET code  
System.Threading.Thread.Sleep(6000)
```

パスワードの暗号化

セキュリティ上の問題を避けるために、テキスト形式のパスワード系文字列を暗号化します。

1. メニューから、**ツール > パスワード エンコーダー** を選択します。 **パスワード エンコーダー** ダイアログが表示されます。
2. **パスワード** フィールドに、暗号化する文字列を入力します。暗号化したパスワードは、**エンコード済み文字列** フィールドに表示されます。
3. **クリップボードにコピーして閉じる** をクリックして、暗号化したパスワードをクリップボードにコピーして、**パスワード エンコーダー** ダイアログを閉じます。
4. ビジュアル テスト、またはテスト スクリプトの該当するコード行を変更します。

 **注:** Desktop Encrypt メソッドを使って、コードで文字列を暗号化することもできます。暗号化した文字列は、Decrypt メソッドを使って復号できます。

 **注:** Silk Test はデフォルトのアルゴリズムを使って暗号化します。使用するアルゴリズムは、%APPDATA%\Silk\Silk Test\conf にある encryption.properties.sample ファイルで algorithm と key をカスタマイズすることにより変更できます。カスタマイズ方法については、プロパティ ファイルのコメントを参照してください。カスタマイズしたアルゴリズムを有効にするには、Silk Test を終了し、トレイ アイコンのメニューから Open Agent をシャットダウンします。そして、プロパティ ファイルの名前を encryption.properties.sample から encryption.properties に変更してから Silk Test を再起動します。

テストのデバッグ

ビジュアルテストおよびスクリプトのデバッグプロセスについて説明します。

ビジュアルテストのデバッグ

再生中のエラーは、テストアプリケーションの変更、ビジュアルテストの不適切なステップフロー、環境の変更など、さまざまな要因により発生します。デバッグ機能を使用してこれらのエラーを迅速に診断、修正することで、ビジュアルテストのメンテナンスの労力を最小限に抑え、チームによるテスト作業の効率を上げることができます。

ビジュアルテストのデバッグを使用すると、ビジュアルテストの再生を一時的に中断して、その再生の管理、検証、リセット、段階的な再生を実行することができます。これにより、エラーの発生箇所を正確に識別できます。デバッグ中にステップを編集できるため、多くの場合、ステップを再実行する前にその中のエラーを解決できます。

選択したポイントでのビジュアルテスト再生の停止

ビジュアルテストの再生中に特定のポイントで停止して、デバッグモードに入るようにビジュアルテストを構成できます。再生中にビジュアルテストが停止するように設定するポイントのことを、ブレークポイントと呼びます。ブレークポイントにより、デバッグしやすいようにビジュアルテストの再生を管理でき、ビジュアルテストが失敗する箇所を切り離すのに役立ちます。ブレークポイントを使用すると、ビジュアルテストにエラーがない場合でも、ビジュアルテストがどのように再生されるかを分析できます。ブレークポイントを設定すると、ブレークポイントを設定したステップを実行する前に再生が停止し、デバッグモードに入ります。

ブレークポイントの設定により、確認したい特定のポイントまでビジュアルテストを再生できるようになります。ブレークポイントは、再生を停止したい特定のステップに設定します。たとえば、変数に含まれるデータの状態をローカル変数ウィンドウで確認したいポイントに設定します。再生は最初のブレークポイントまで実行され、ビジュアルテストに続行を指示するまで中断されます。

1. ブレークポイントを設定するビジュアルテストを開きます。
2. 再生を停止するステップを選択します。
3. **デバッグ > ブレークポイントの設定/クリア** を選択します。ステップ番号の横の列に、ブレークポイントアイコンが表示されます。再生は、設定されたブレークポイントまで実行されます。再生がブレークポイントで停止すると、ビジュアルテストのブレークポイントステップが黄色で強調表示され、そのステップを指す黄色の矢印が表示されます。

ビジュアルテストがデバッグモードで表示され、再生を段階的に実行したり、ステップの再生を制御したり、再生をビジュアルテスト内の特定のポイントで開始または停止したりできるようになります。



ヒント: ブレークポイントが設定されたステップを選択し、**F9** を押すと、選択したステップからブレークポイントが削除されます。**Ctrl+Shift+F9** を押すと、すべての開いているビジュアルテストとVB.NET スクリプトのすべてのブレークポイントが削除されます。

選択したポイントからのビジュアルテスト再生の段階的実行

デバッグ中に、ブレークポイントによって再生が一時停止された場所から、1行ごとにビジュアルテストを実行できます。

1. ビジュアル テストの特定のステップにブレークポイントを設定します。
2. F5 を押して、ビジュアル テストを再生します。再生は、ブレークポイントまで実行されると停止し、デバッグ モードになります。**テスト ステップ** ペインで、次に再生されるステップが黄色で強調表示されます。
3. F8 を押して、強調表示されたステップを再生します。
ステップが正常に実行されたら、再生は再びデバッグ モードになり、次に再生されるステップが黄色で強調表示されます。ステップが正常に再生されなかった場合は、**再生エラー** ダイアログ ボックスが開きます。
4. F8 を押して、ビジュアル テスト内の残りのステップの再生を段階的に実行します。再生は、設定されたブレークポイントまで実行されます。ビジュアル テストがデバッグ モードで表示され、再生を段階的に実行したり、ステップの再生を制御したり、再生をビジュアル テスト内の特定のポイントで開始または停止したりできるようになります。ビジュアル テストの段階的実行中、ローカル変数 ウィンドウを使用して、テスト実行の重要なポイントにおける変数の使用状況をスナップショットとして取得できます。デバッグ中に、特定のステップから再生を開始するように、ビジュアル テストを構成することもできます。

ビジュアル テストの再生におけるステップ実行の制御

ビジュアル テストのデバッグ中に、またはブレークポイントで、ビジュアル テスト内で次に実行するステップを制御できます。実行ポイントを移動すると、他のステップを実行せずに、ビジュアル テスト内の別の箇所にある選択したステップから再生を再開できます。

実行ポイント/次のステートメントの設定 コマンドは、選択したステップに実行ポイントを設定します。この機能は、デバッグ中のブレークポイントでのみ使用可能です。

現在のプロシージャ内でステートメントを再実行する場合や、ビジュアル テスト内で再生したくないステップをスキップする場合は、**実行ポイント/次のステートメントの設定** コマンドを使用します。この機能を使用すると、ビジュアル テストのデバッグ中にフローを制御してエラーを認識しやすくできるため、エラーの診断と修正に要する時間を短縮することができます。

1. デバッグ中、または再生中のブレークポイントで、実行する次のステップを選択します。
ビジュアル テストで任意のステップを選択します。
2. **デバッグ > 実行ポイント/次のステートメントの設定** を選択します。選択したステップに黄色の実行矢印が移動し、そのステップが黄色で強調表示されます。
3. F5 を押して選択したステップから再生を再開するか、F8 を押して選択したステップのみを再生します。選択したステップが実行されると、再生が停止し、Silk Test Workbench がデバッグ モードに戻ります。実行ポイントは次のステップに移動します。

デバッグ中のビジュアル テストの編集

デバッグ中にビジュアル テストを編集および保存することができます。変更を保存すると、そのビジュアル テストの新しいバージョンが作成され、アクティブ ビジュアル テストになります。

デバッグ中にビジュアル テストを段階的に実行して、変更を加えることができます。加えた変更は、再生時に適用されます。デバッグ中に、ビジュアル テストで以下の変更を実行することができます。

- フラグの追加。
- 既存のステップのプロパティの編集。
- テストステップのコピーと貼り付け。
- ブレークポイントの設定と解除。
- 他のビジュアル テストに挿入されるビジュアル テストの編集。たとえば、*visualtest2* を *visualtest1* に挿入する場合、*visualtest2* を制限なしで編集できます。

デバッグモードでのビジュアルテストの段階的実行

デバッグモードでは、ビジュアルテストを1ステップずつ再生できます。ステップングと呼ばれるこの方法により、再生中にステップを段階的にトレースして、テストステップの実行順序、特定のステップによって返されるデータ、およびステップが正常に実行されるかどうかを確認できます。

デバッグモードでは、以下のコマンドを使用して再生の実行を制御します。

ステップ イン (F8) 1つずつステップを再生します。**ステップ イン** は各ステップのトレースに有効であり、また再生しているビジュアルテストに挿入された別のビジュアルテストやスクリプトにもステップ インします。挿入された各ビジュアルテストやスクリプトについても、一度に1ステップまたは1行ずつ実行されます。

ステップ イン はテストの詳細分析に有効であり、変数の使用やターゲットアプリケーションとの対話に対する各ステップの影響を確認することができます。

ステップ オーバー (Shift + F8) 別のビジュアルテストに挿入されたビジュアルテスト全体またはスクリプト全体を、1つのステップとして実行します。**ステップ オーバー** は、デバッグモードの再生中に、ビジュアルテストまたはスクリプトを1ステップとして再生する場合に使用します。これにより、挿入されたビジュアルテスト全体またはスクリプト全体が再生されます。挿入されたビジュアルテスト全体またはスクリプト全体が再生されると、元のビジュアルテスト内の次のステップで再生が停止し、デバッグモードになります。

別のビジュアルテストまたはスクリプトを再生するステップ以外のステップで**ステップ オーバー**を使用すると、**ステップ イン**を使用した場合と同じになります。次のステップのみが実行されたあと、再生は停止し、再びデバッグモードになります。

ステップ アウト (Ctrl + Shift + F8) 別のビジュアルテストから再生しているビジュアルテストまたはスクリプト内の残りのステップをすべて実行したあと、元のビジュアルテスト内の次のステップで再生が一時停止します。

ステップ アウト は、別のビジュアルテストに挿入されたビジュアルテスト内のステップまたはコード行で再生が停止されているときに、残りのビジュアルテストを再生して元のビジュアルテストに戻る場合に使用します。挿入されたビジュアルテストまたはスクリプトの残りの部分の再生が実行されたら、元のビジュアルテスト内の次の行で再生が停止し、再びデバッグモードになります。

カーソルまで実行 (Ctrl + F8) 再生を一時停止するステップを選択できます。これにより、ビジュアルテストまたはスクリプト内の選択したセクションを「ステップ オーバー」することができます。

カーソルまで実行 は、ビジュアルテストまたはスクリプトを再生して、実行時エラーが発生したポイントの直前で停止する場合に使用します。これにより、ブレークポイントを挿入することなく特定の行またはステートメントで再生を停止できます。再生が停止したら、別のデバッグオプションのいずれかを使用して再生を継続できます。

カーソルから実行 現在選択されているテストステップからビジュアルテストを再生します。

ステップングコマンドには、**デバッグ**メニューからアクセスします。

ビジュアルテスト内でのエラー処理による確実な再生

アプリケーションのテストは、テスト対象アプリケーション内のエラーを明らかにするものですが、Silk Test Workbenchを使用したソフトウェアアプリケーションのテストでは、これらのエラーが原因でビジュアルテストの再生を完了できない場合があります。たとえば、テストステップで、再生中に検出されないTextFieldにテキストを入力しようとする、オブジェクトが見つからないというエラーが発生し、再生が停止します。ユーザーはデバッグモードに切り替えて、エラーの原因を特定することができます。

エラー処理を設定すると、再生エラーをデバッグする必要がなくなります。エラー処理を設定するには、ビジュアルテストでエラーの検出ステップとその処理方法を設定します。エラー処理を設定することによって、ビジュアルテストを最後まで再生でき、再生中に発生したエラーの情報を得ることができます。

[プロパティ] ペインを使用して、エラーに自動的に対処するようにビジュアルテストを設定

ビジュアルテストは、再生エラーを予測して自動的に対処するように構成することができます。ビジュアルテストのステップ内でエラーを処理するように設定するか、特定のエラーを処理するステップを含む別のビジュアルテストを開くように設定できます。エラーを処理するようにビジュアルテストを設定するには、エラー処理ステップを作成します。

1. エラー処理を設定するビジュアルテストを開きます。
2. エラー処理ステップを作成する場所の直前のステップを選択します。
3. **挿入 > エラー処理** を選択します。選択したステップのあとにエラー処理ステップが作成されます。エラー処理ステップのプロパティが**プロパティ** ペインに表示されます。
4. 新しく作成されたステップを選択して、**プロパティ** ペインにそのプロパティを表示します。
5. エラー処理ステップの**プロパティ** ペインでツールバー ボタンをクリックして、プロパティをカテゴリ別にグループ化するか、アルファベット順に並べ替えます。
6. **エラー時の移動先** プロパティを更新し、再生設定とその値を割り当てます。
エラーハンドラの**実行する操作** プロパティが**移動**である場合、ビジュアルテスト内でエラーハンドラの移動先になるラベルを指定します。

エラー処理ステップは、再生中にエラーが発生すると実行されます。

テストロジックデザイナーを使用してビジュアルテストを設定し、エラーに自動的に対処

ビジュアルテストは、再生エラーを予測して自動的に対処するように構成することができます。ビジュアルテストのステップ内でエラーを処理するように設定するか、特定のエラーを処理するステップを含む別のビジュアルテストを開くように設定できます。ビジュアルテストでエラーが処理されるように設定するには、**テストロジックデザイナー** ウィザードを使用します。

1. エラー処理を設定するビジュアルテストを開きます。
2. エラー処理ステップを作成する場所の直前のステップを選択します。
3. **挿入 > テストロジック > エラー処理** を選択します。**テストロジックデザイナー** ウィザードが開き、**ようこそ** ページが表示されます。
4. **次へ** をクリックします。**ロジックの種類を選択** ページが開きます。
5. **ビジュアルテストの再生中のエラー (再生エラー)** をクリックし、**次へ** をクリックします。**エラー処理の構築** ページが開きます。
6. **再生エラーが発生した場合、常に以下の操作を行う** リストで、以下のいずれかの操作を選択します。

デフォルトの再生エラーダイアログを表示します **再生エラー** ダイアログボックスを表示し、そこでそのエラーに対して実行する操作を決定します。**再生エラー** ダイアログボックスを使用してデバッグモードに入り、ビジュアルテストでエラーを診断します。

現在のビジュアルテストの再生を完了します エラーが発生したときに、ビジュアルテストの再生を終了します。ビジュアルテストのその他のステップは再生されません。再生が終了すると、**再生完了** ダイアログボックスが表示され、そこで再生の完了時に実行する操作を決定します。

エラーが発生したステップの再実行 **資産** プロパティに指定されているビジュアルテストを再生してから、エラーが発生したステップの実行を試行します。**資産** プロパティにビジュアルテストが指定されていない場合、単にエラーが発生したステップの再実行を試行します。**再試行の回数**

- 生を再試行します** プロパティには、再生を試行する回数を指定できます。デフォルト値は、5 です。指定した再試行の回数に到達した時点でステップ実行に失敗すると、再生エラーが生成されます。
- 次のステップで再生を再開します** **資産** プロパティに指定されているビジュアル テストを再生してから、エラーが発生したステップの直後のステップの実行を試みます。エラーの原因となったステップは、再実行されません。
- ラベルへ移動します** **資産** プロパティに指定されているビジュアル テストを再生し、**ラベル** プロパティに指定されている事前定義のラベル ステップに移動して、再生を続行します。**移動** を使用すると、ビジュアル テストで、再生中に発生したエラーと同様のエラーが発生する可能性のあるステップをスキップして、テストの別のセクションに移動できます。
- 他のビジュアル テストを最初に再生します** 指定した別のビジュアル テストが再生されたあと、次に実行される操作をユーザーが指定します。別のテスト実行のあとに実行される操作を決定するには、テストの終了、エラーが発生したステップを再試行、ラベルへ移動、またはエラーが発生した元のテストの再開を選択します。

7. **ラベルへ移動します** を指定した場合は、**移動先のラベルを選択する** リスト ボックスから エラー ハンドラが移動するラベルを選択します。
8. **他のビジュアル テストを最初に再生します** を指定した場合は、以下のステップを実行します。
 - a) **<ビジュアル テストの挿入>** をクリックし、再生する**資産**を選択して、**OK** をクリックします。
 - b) **次の処理** リスト ボックスから、次に行う操作を選択します。
 - ラベルへ移動します** を指定した場合は、**移動先のラベルを選択する** リスト ボックスから エラー ハンドラが移動するラベルを選択します。
9. **次へ** をクリックします。 **要約** ページに、新しいエラー処理ステップのテスト ステップの説明が表示されます。
- 10 **終了** をクリックします。

ビジュアル テストの印刷

ビジュアル テストを印刷して、テストの内容を保存します。

1. 印刷するビジュアル テストを開きます。
2. **ファイル > 印刷** を選択します。**印刷** ダイアログ ボックスが開きます。
3. リストからプリンタを選択します。
4. 省略可能：**プロパティ** をクリックすると、余白や印刷の向きなどの印刷ページ設定を変更できます。
5. 省略可能：**印刷範囲** セクションで、ドキュメント全体を印刷するか、または選択したページのみを印刷するかを指定します。
6. 省略可能：**部数** フィールドで、印刷する部数を指定します。
7. 省略可能：**部単位で印刷する** チェック ボックスをオンにすると、すべてのコピーの対応するページをまとめて印刷します。
 - このチェック ボックスは、複数の部数を印刷するように選択した場合にのみ有効になります。
8. 印刷するステップを指定します。
 - **テスト ステップ** ペインに表示されているすべてのステップを印刷する場合は、**現在の表示** をクリックします。
 - **テスト ステップ** ペインで選択されているステップだけを印刷する場合は、**選択** をクリックします。
9. 印刷出力をカスタマイズするには、**オプション** をクリックして、印刷出力に含める情報の種類を選択します。
 - 以下のオプションがあります。

- 以下を含める** 含める内容の種類を指定します。
- **ビジュアルテストの要約**：ビジュアルテストの名前、プロジェクト、作成者、作成日、バージョン、説明などのビジュアルテストの要約を含めます。この要約は、印刷出力の最初のページに表示されます。
 - **最新の結果の要約**：ビジュアルテストの最新結果の要約ページを含めます。この要約は、印刷出力の最後のページに表示されます。
 - **グループ画面**：現在のダイアログボックスまたは Web ページのメインコンテンツを含むグループ画面のスナップショットを印刷出力に含めます。グループ画面に続くテストステップは、現在のダイアログボックスまたは Web ページのコンテンツで発生する単一の操作に対応します。グループ画面は、**テストステップ** ペインの Using ステップに対応します。グループ画面のスナップショットは、**画面プレビュー** ペインで表示されるように、印刷出力に示されます。
 - **無効なステップ**：無効化されたステップを含めます。この情報は、印刷出力で灰色のテキストで表示されます。

- 各ステップに含める** 各ステップに含める内容を指定します。
- **画面**：各ステップの関連画面スナップショットを含めます。個々の画面は (ステップ画面) に対応し、現在のダイアログボックスまたは Web ページでの特定の操作に関連付けられます。ステップ画面のスナップショットは、**画面プレビュー** ペインで表示されるように、印刷出力に示されます。
 - **説明**：各ステップの説明を含めます。ステップの説明は、**プロパティ** ペインの **ステップの説明** プロパティに表示されるように、印刷出力に示されます。
 - **フラグの詳細**：フラグの説明、作成者、更新者、更新日、割り当てられたユーザーなどのフラグの詳細を含めます。

- 以下のプロパティを含める** 各ステップに含めるプロパティを指定します。
- **すべてのプロパティ**：各ステップのすべてのプロパティを含めます。

デフォルト デフォルトの印刷オプションを選択するには、このボタンをクリックします。

- 10**省略可能 : **プレビュー** をクリックして、結果情報がどのように印刷されるかを表示します。**プレビュー** ウィンドウが開きます。
- 11****OK** をクリックします。

スクリプトのデバッグ

再生中の実行時エラーは、さまざまな要因により発生します。これらの要因には、テストアプリケーションの変更、テストスクリプトのコーディングの誤り、または環境の変更などがあります。スクリプトのデバッグによってこれらのエラーを迅速に診断、修正することで、スクリプトのメンテナンスの労力を最小限に抑え、より効率的なチームによるテスト作業を推進することができます。

スクリプトのデバッグを使用すると、開発環境でのスクリプトの再生を一時的に中断して、その再生の管理、検証、リセット、段階的実行、継続を行えます。

Silk Test Workbench には **出力** ウィンドウがあり、自動処理コードの確認やユーザー定義の追加出力の追加を行うことができます。たとえば、`Console.Write` および `Console.WriteLine` コマンドを使用して、追加実行を出力できます。

さらに、ローカル変数 ウィンドウを使用してドリルダウンし、スクリプト内のすべての変数の値を表示することもできます。

スクリプトの信頼性

アプリケーションのテストは、テスト対象アプリケーション内のエラーを明らかにします。ソフトウェアアプリケーションのテスト時に、これらのエラーが原因でスクリプトの実行を完了できない場合があります。

テストを成功させるには、スクリプトの再生を確実に完了させる必要があります。同期化の問題、あるいは予期されるまたは予期されないアプリケーション エラーによって、再生が完了する前にスクリプトが失敗することがあります。Silk Test Workbench に組み込まれているオプション設定とスクリプト作成機能により、同期化とアプリケーションの問題を調整して、任意のテスト環境でスクリプトを確実に再生できるようにします。

スクリプトを再生しているときに、アプリケーションの応答時間の遅延に起因するエラーが発生すると、アプリケーションとそのアプリケーションをテストするスクリプトとの同期が取れなくなります。これらのエラーは、Silk Test Workbench のオプション設定を調整してスクリプトとテスト対象アプリケーション間の同期を最適化することで処理できます。

デバッグ モードでのスクリプトの段階的実行

デバッグ モードでは、スクリプトを一度に 1 行ずつ再生できます。ステップングと呼ばれるこの方法により、テスト担当者は再生中にコードを段階的にトレースして、ステートメントの実行順序を確認できるため、デバッグに役立ちます。

ステップングにより、デバッグ モードでスクリプトを段階的に実行し、その実行を詳細に検証することができます。

デバッグ モードでは、以下のコマンドを使用して再生の実行を制御します。

ステップ イン (F8)

一度に 1 ステートメントずつスクリプトを実行します。**ステップ イン** はコードの各行のトレースに有効であり、関数または埋め込まれているスクリプトにもステップ インします。各関数または埋め込みスクリプトも、一度に 1 行ずつ実行されます。

ステップ イン はテストの詳細分析に有効であり、変数の使用やターゲット アプリケーションとの対話に対する各ステートメントの影響を確認することができます。



注: VB .NET スクリプトのデバッグ中に反復子 (*Iterator*) に**ステップ イン**することはできません。反復子は実行されますが、デバッグは実行時に反復子が処理されていることを表示しません。反復子の内側にブレイクポイントを設定すると、実行はブレイクポイントで停止しますが、反復子内のパラメータやローカル変数は、**ローカル変数** ウィンドウに表示されません。詳細については、「[反復子](#)」を参照してください。

ステップ オーバー (Shift + F8)

各プロシージャを 1 つのステートメントとして実行します。関数を構成する各行を実行せずに、関数に直接移動する場合には、[ステップ イン]ではなくこのコマンドを使用します。

ステップ アウト (Ctrl + Shift + F8)

プロシージャ内の残りすべてのコードを 1 つのステートメントのように実行し、そのプロシージャを最初に呼び出したプロシージャ内の次のステートメントに移ります。

カーソルまで実行 (Ctrl + F8)

再生を一時停止するステップを選択できます。これにより、ビジュアル テストまたはスクリプト内の選択したセクションを「ステップ オーバー」することができます。

カーソルまで実行 は、ビジュアル テストまたはスクリプトを再生して、実行時エラーが発生したポイントの直前で停止する場合に使用します。これにより、ブレイクポイントを挿入することなく特定の行またはステートメントで再生を停止できます。再生が停止したら、別のデバッグ オプションのいずれかを使用して再生を継続できます。

実行ポイント/次のステートメントの設定 (Ctrl + F9) 現在ポイントがある場所のステートメントから、途中のステートメントを実行せずステートメントの に実行を続けます。

ステップング コマンドには、**デバッグ** メニューからアクセスします。

選択したポイントからのスクリプト再生の段階的実行

デバッグ中に、ブレークポイントによって再生が一時停止された場所から、スクリプトを実行できます。

1. スクリプトの特定の行にブレークポイントを設定します。
2. F5 を押して、スクリプトを再生します。再生は、ブレークポイントまで実行されると停止し、デバッグモードになります。
3. F8 を押して、スクリプトを次のブレークポイントまで再生します。
コードが正常に実行されたら、再生は再びデバッグ モードになり、次に再生されるコードが黄色で強調表示されます。コードが正常に再生されなかった場合は、**再生エラー** ダイアログ ボックスが開きます。
4. F8 を押して、スクリプト内の残りのコードの再生を段階的に実行します。再生は、設定されたブレークポイントまで実行されます。スクリプトがデバッグ モードで表示され、再生を段階的に実行したり、ステップの再生を制御したり、再生をスクリプト内の特定のポイントで開始または停止したりできるようになります。スクリプトの段階的実行中、**変数の監視** ウィンドウを使用して、テスト実行の重要なポイントにおける変数の使用状況をスナップショットとして取得できます。デバッグ中に、コード内の特定の行から再生を開始するように、スクリプトを構成することもできます。


スクリプト デバッグ中の行の実行の制御

スクリプトのデバッグ中にフローを制御して、エラーを認識しやすくし、エラーの診断と修正に要する時間を短縮することができます。

実行ポイント/次のステートメントの設定 コマンドは、スクリプト内の選択した行に実行ポイントを設定します。この機能は、デバッグ中のブレークポイントでのみ使用可能です。

現在のプロシージャ内でステートメントを再実行する場合や、スクリプト内で再生したくないステートメントをスキップする場合は、**実行ポイント/次のステートメントの設定** コマンドを使用します。この機能を使用すると、スクリプトのデバッグ中にフローを制御してエラーを認識しやすくできるため、エラーの診断と修正に要する時間を短縮することができます。

1. デバッグ中、または再生中のブレークポイントで、以下のいずれかのステップを実行します。
 - 黄色い矢印をクリックして次に再生するコード行へドラッグします。
 - 次に再生する行をクリックしてカーソルを配置してから、**デバッグ > 実行ポイント/次のステートメントの設定** を選択します。

 **注:** スクリプトで、次のステートメントは、現在のステートメントと同じプロシージャの行にのみ設定できます。
2. F5 を押して選択した行から再生を再開するか、F8 を押して選択した行のみを再生します。選択したステップが実行されると、再生が停止し、Silk Test Workbench がデバッグ モードに戻ります。実行ポイントは次の行に移動します。

スクリプト例 - エラー処理

以下のスクリプトは、エラー処理を含むスクリプトを作成する方法を示しています。

基本的なエラー処理をスクリプトに含めるには、Catch ex As Exception ステートメントに続いて例外のテスト対象となるオブジェクトとパラメータをスクリプトに含めます。たとえば、メモ帳アプリケーションのフォントの種類をテストするには、以下のコードを入力します。

```
Public Module Main
    Dim _desktop As Desktop = Agent.Desktop
```

```

Public Sub Main()
    With _desktop.Window("@caption='Untitled - Notepad'")
        .TextField().TypeKeys("test", 0, False)
        .MenuItem("@caption='Font']").Select()
    Try
        With .Dialog("@caption='Font1'") ' throws an exception because such a
dialog does not exist
            .ComboBox("@caption='Font style:']").Select("Regular")
            .ComboBox("@caption='Size:']").Select("14")
            .PushButton("@caption='Cancel']").Select()
        End With
    Catch ex As Exception
        With .Dialog("@caption='Font'")
            .ComboBox("@caption='Font style:']").Select("Regular")
            .ComboBox("@caption='Size:']").Select("14")
            .PushButton("@caption='Cancel']").Select()
        End With
    End Try
End With
End Sub
End Module

```



注: この例は、基本的なエラー処理の実装方法を示すことを意図しています。このコードを正しく実装するには、アプリケーション構成を追加して、テスト アプリケーションを必要に応じて調整する必要があります。このサンプル コードをスクリプトに貼り付けても、アプリケーション構成が存在しないため、エラーが発生します。

スクリプトへのコメントの挿入

出力 ウィンドウを使用して、自動処理コードを確認し、ユーザー定義の追加出力を付加して、スクリプトに関する補足情報を提供します。たとえば、コメントを追加して、スクリプトに関する補足情報を提供できます。スクリプトを再生すると、コメントが **出力** ウィンドウに表示されます。

1. スクリプトでは、`Console.Write` および `Console.WriteLine` コマンドを使用してコメントを追加できます。
`Comment` メソッドでは、文字列がパラメータとして処理されます。この文字列の内容は、スクリプトの再生後に **出力** ウィンドウに保存されます。
2. スクリプトを再生します。 **出力** ウィンドウに、自動処理コードおよび追加したコメントを表示できます。

選択したポイントでのスクリプト再生の停止

スクリプトの再生中に特定のポイントで停止して、デバッグ モードに入るようにスクリプトを構成できます。再生中にスクリプトが停止するように設定するポイントのことを、ブレークポイントと呼びます。ブレークポイントにより、デバッグしやすいように再生を管理でき、スクリプトが失敗する箇所を切り分けるのに役立ちます。ブレークポイントを使用すると、スクリプトにエラーがない場合でも、スクリプトがどのように再生されるかを分析できます。ブレークポイントを設定すると、再生はブレークポイントで停止し、デバッグ モードに入ります。

ブレークポイントの設定により、確認したい特定のポイントまでスクリプトを再生できるようになります。ブレークポイントは、再生を停止したい特定の行に設定します。たとえば、変数に含まれるデータの状態を **変数の監視** ウィンドウで確認したいポイントに設定します。再生は最初のブレークポイントまで実行され、スクリプトに続行を指示するまで中断されます。

1. ブレークポイントを設定するスクリプトを開きます。

- 再生を停止する行をクリックします。
- デバッグ > ブレークポイントの設定/クリア**. 行番号の横の列に、ブレークポイントアイコンが表示されます。再生は、設定されたブレークポイントまで実行されます。再生がブレークポイントで停止すると、スクリプトのブレークポイント行が黄色で強調表示され、その行を指す黄色の矢印が表示されます。

デバッグでは、再生中に行の実行を制御できます。また、特定の行まで再生を実行したり、特定の行から再生を開始したりするように、スクリプトを構成することもできます。



ヒント: ブレークポイントが設定された行を選択し、F9 を押すと、選択した行からブレークポイントが削除されます。Ctrl+Shift+F9 を押すと、スクリプトからすべてのブレークポイントが削除されます。

スクリプトの印刷

確認やトラブルシューティングに役立つために、コード ウィンドウに表示されているコードを印刷できます。

- 印刷するスクリプトを開きます。
- ファイル > 印刷** を選択します。**印刷** ダイアログ ボックスが開きます。
- リストからプリンターを選択するか、**プリンター検索** をクリックして他のプリンターを検索します。
- 省略可能：**ファイルへ出力** チェック ボックスをオンにすると、PDF などのファイルへ出力されます。
- 省略可能：**詳細設定** をクリックすると、余白や印刷の向きなどの印刷ページ設定を変更できます。
- 省略可能：**印刷範囲** セクションで、ドキュメント全体を印刷するか、または選択したページのみを印刷するかを指定します。
以下のいずれかのオプションをクリックします。
 - すべて**：ドキュメント全体を印刷します。
 - 選択した部分**：選択したコードを印刷します。
 - ページ指定**：指定したページを印刷します。
 - 現在のページ**：現在のページを印刷します。
- 省略可能：**部数** フィールドで、印刷する部数を指定します。
- 省略可能：**部単位で印刷する** チェック ボックスをオンにすると、すべてのコピーの対応するページをまとめて印刷します。
このチェック ボックスは、複数の部数を印刷するように選択した場合にのみ有効になります。
- 印刷** をクリックします。

テストの再生と結果の分析

テストの再生および再生結果の分析の手順について説明します。

ビジュアル テストの再生

ビジュアル テストを記録したら、いつでもそれを再生できます。テスト ステップは、上から下に順番に実行されます。再生中、ビジュアル テストは、記録または追加された操作に従ってテスト アプリケーションを操作します。これらの操作は画面で確認できます。操作は、そのアプリケーションで可能な最短の応答速度で実行されます。ビジュアル テストは、手動での実行に比べはるかに短時間で再生できます。

ビジュアル テストは、オブジェクトの画面上の位置に関係なくオブジェクト レベルで記録されるため、再生もオブジェクト指向です。つまり、Silk Test Workbench はオブジェクトと直接対話します。そのため、環境やテスト アプリケーションに変更があった場合でも、ビジュアル テストは正しく再生されます。

ビジュアル テストを再生するたびに、そのテストに記録されたすべての操作が Silk Test Workbench によって繰り返され、検証などの記録または追加された操作が実行されます。

ビジュアル テストの再生

ビジュアル テストを再生すると、記録または追加した操作が実行されます。

1. 次のいずれか 1 つのステップを行います：

- **開始画面** の **資産** セクションの **最新** リストで、再生するビジュアル テストの **ビジュアル テスト 'プロジェクト名 - テスト名'** の再生 をクリックします。
- **アセットブラウザ** で、再生するビジュアル テストを右クリックして、**再生** をクリックします。
- **ビジュアル ナビゲータ** で **操作 > 再生** を選択し、アクティブなビジュアル テストを再生します。

再生 ダイアログ ボックスが開きます。

2. 省略可能：テスト結果の説明を **結果の説明** フィールドに入力します。

3. **結果オプション** リストから、このテストの結果の保存方法を選択します。

以下のオプションのいずれかを選択します。

新しい実行番号で結果を保存 ビジュアル テストまたはスクリプトの再生の結果は、新しい増分実行番号で保存されます。たとえば、新規の結果は実行番号 1 で保存されます。同じビジュアル テストまたはスクリプトの後続の結果は、実行番号 2、3、4 など保存されます。これはデフォルトの設定です。

現在の実行番号に結果を追加 現在の再生の結果は、最新の実行番号の結果に付加されます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。

既存の実行番号を新しい結果で上書き 現在の再生の結果で、最新の既存の実行番号の結果が上書きされます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。

この結果を保存しない ビジュアル テストまたはスクリプトを再生したときに結果は作成されません。

4. 省略可能：**オプション** をクリックして、結果オプションを変更します。

詳細については、「再生結果オプションの設定」を参照してください。

5. 省略可能：**ローカル変数ウィンドウの表示** チェック ボックスをオンにすると、再生中に **ローカル変数** ウィンドウが表示されます。

Silk Test Workbench は、再生が終わると、**ローカル変数** ウィンドウを非表示にします。

6. **OK** をクリックします。
7. Web アプリケーションをテストする場合は、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。
8. 省略可能：スクリプトを停止する場合は、**操作 > 停止** を選択するか、ツールバーの **停止** をクリックするか、両方の **Shift** キーを同時に押します。現在のステップで再生を一時停止する場合は、**操作 > 一時停止** を選択するか、ツールバーの **一時停止** をクリックするか、**Alt + F11** キーを押します。ビジュアル テストが再生されるときに、テスト アプリケーションに対して実行される各操作を確認できます。

ビジュアル テスト内に値が割り当てられていない受け渡し変数が見つかったら、**ビジュアル テスト変数** ダイアログ ボックスが表示され、これらの変数に値を割り当てることができます。

再生中にエラーが発生すると、**再生エラー** ダイアログ ボックスが開きます。

9. 再生が完了したら、**再生完了** ダイアログ ボックスを使用して、実行する操作を決定します。



注: 再生中にエラーが発生した場合は、エラー メッセージが表示されます。

再生中、**再生** ダイアログ ボックスを非表示にするには、**このダイアログを再表示しない** チェック ボックスをオンにします。これにより、全般オプションの **[再生] ダイアログの表示** オプション設定が上書きされます。**再生** ダイアログ ボックスを表示するには、この設定を使用します。

ビジュアル テスト内でのビジュアル テストの実行

ビジュアル テストでは、別のビジュアル テストを呼び出して実行したり、再生したりできます。これにより、アプリケーションのテストに対するモジュール式アプローチが可能になり、テストの実行をより制御できるようになります。ビジュアル テスト内で別のビジュアル テストを実行する場合は、テスト サイトの整合性を保ち、テスト対象のターゲット アプリケーションが正しい初期再生状態になるようにすることが重要です。

この手順を使用して、ドライバ テストを作成することもできます。

1. 別のビジュアル テストを挿入するビジュアル テストを開きます。
2. テストの挿入箇所の直後にあるステップを選択します。
3. **挿入 > ビジュアル テスト** をクリックします。 **ビジュアル テストの参照** ダイアログ ボックスが開きます。
4. すべての資産を表示するか、作成した資産だけを表示するかを選択します。



注: 列見出し (**名前**、**プロジェクト**、**説明**、**更新日**) をクリックして、その内容を基準にビジュアル テストのリストをアルファベット順に並べ替えることができます。

5. 挿入するビジュアル テストを選択し、**OK** をクリックします。



注: 現在のプロジェクトとすべての参照プロジェクトのビジュアル テストがリストに表示されません。



ヒント: **資産の選択** リストに多くのビジュアル テストが表示されている場合、**名前フィルタ** テキスト フィールドに挿入したいビジュアル テストの名前を入力してから、ビジュアル テストをクリックすると簡単です。

Silk Test Workbench によって、選択したステップの下にステップが挿入されます。挿入されたステップによって、選択したビジュアル テストが呼び出されます。


再生中、上記のステップが実行されると、呼び出し元のビジュアル テスト内の次のステップが実行される前に、挿入されたビジュアル テストが最後まで再生されます。ビジュアル テストが別のビジュアル テストに挿入されたあと、再生に使用される設定を構成できます。

再生すると、テスト対象アプリケーションは、必要に応じて設定、フック、開始され、再生セッションの終了時までフックされたままになります。たとえば、呼び出し元のビジュアル テストにメモ帳用のアプリケーション構成があり、挿入されたビジュアル テストに電卓用のアプリケーション構成がある場合には、次のようになります。

- 呼び出し元のビジュアル テストの実行の開始時に、メモ帳がフックされ、開始されます。
- 挿入されたビジュアル テストの実行の開始時に、電卓がフックされ、開始されます。メモ帳はフックされたままです。
- 挿入されたビジュアル テストの実行を停止しても、メモ帳と電卓の両方ともフックされたままになります。
- 再生セッションの終了時、つまり、呼び出し元のビジュアル テストの実行を停止すると、メモ帳と電卓の両方のフックが解除されます。

[ビジュアル テスト変数] ダイアログ ボックス

このダイアログ ボックスを使用すると、再生中のビジュアル テストで値が割り当てられていない変数が渡された場合に、値を割り当てることができます。このダイアログ ボックスは、再生中に、ビジュアル テストの <<開始>> ステップで定義されたパラメータまたは変数に値がない場合にのみ表示されます。

 **注:** 変数名は、「st_」で始めることはできません。


ビジュアル テスト変数 ダイアログ ボックスには、以下の列で構成されるリストが含まれています。

ビジュアル テスト変数名: このビジュアル テスト内の値が割り当てられていないビジュアル テスト変数の名前が表示されます。この変数は、再生中にこのビジュアル テストを実行する他のビジュアル テストにも渡されます。この列のデータは表示専用で変更はできません。

型: ビジュアル テスト変数のデータ型が表示されます。この列のデータは表示専用で変更はできません。変数の型を以下に示します。

テキスト

変数の値はテキスト文字列です。テキスト型の値には、文字、数字、スペース、および句読点が含まれます。

 **注:** テキスト データ型は大文字/小文字を区別します。つまり、比較時にはテキスト文字列の個々の文字の大文字/小文字が一致していなければなりません。大文字/小文字が一致していない場合は、比較は失敗します。

数値 (Long)

変数の値は、-2,147,483,648 から 2,147,483,647 の範囲の数値です。(小数値または 10 進数値で使用される) ピリオドは使用できません。

数値 (Double)

変数の値は、倍精度浮動小数点数で、-1.7E308 から +1.7E-307 の 64 ビットの数値として格納されます。

ブール値 (True/False)

変数の値は True または False です。

数値 (Long Long)

64 ビット整数値 (-9223372036854775808 から 9223372036854775807 の範囲) です。

列挙

この変数は一連の値をグループ化し、1 から n まで順番に番号を付けます。限られた個数の一意の値のみを変数に保持する場合は、列挙型を宣言します。この変数型は、列挙データ型を必要とするプロパティまたは変数に使用します。

列挙には、64 ビット整数 (-9223372036854775808 から 9223372036854775807 の範囲) が格納されます。ただし、UI では 64 ビットの列挙がサポートされていますが、エンジンでは 32 ビットの列挙のみがサポートされています (将来のリリースでは、エンジンでも 64 ビットの列挙がサポートされる予定です)。

値: リストされたビジュアル テスト変数の値です。ビジュアル テスト変数の値を入力します。値は型に一致している必要があります。たとえば、型が数値 (Long) の場合、値は -2,147,483,648 から 2,147,483,647 の数値にする必要があります。入力した値が型と一致しない場合、または値を入力しなかった場合は、デフォルト値が使用されます。

- 数値 (Long)、数値 (Double)、数値 (Long Long)、および列挙の場合、デフォルト値は 0 です。

- テキストの場合、デフォルト値は NULL 文字列です。
- ブール値の場合、デフォルト値は False です。

スクリプトの再生

スクリプトを記録したら、いつでもそれを再生できます。通常、スクリプト内のコード行は上から下の順に実行されます。スクリプトの再生中、スクリプト内の操作に従ってテスト アプリケーションが操作されます。これらの操作は画面で確認できます。操作は、そのアプリケーションで可能な最短の応答速度で実行されます。これにより、手動での実行に比べはるかに短時間でテストを実行できます。

スクリプトは、オブジェクトの画面上の位置に関係なくオブジェクト レベルで記録されるため、再生もオブジェクト指向です。つまり、Silk Test Workbench は Windows のオブジェクトと直接対話します。そのため、環境やテスト アプリケーションに変更があった場合でも、スクリプトは正しく再生されます。

スクリプトの再生

スクリプトを再生すると、記録または記述された各操作が実行されます。

1. スクリプトを開きます。
2. **操作 > 再生** を選択するか、ツールバーの **再生** をクリックします。 **再生** ダイアログ ボックスが開きます。
3. 省略可能：テスト結果の説明を **結果の説明** フィールドに入力します。
4. **結果オプション** リストから、このテストの結果の保存方法を選択します。

以下のオプションのいずれかを選択します。


新しい実行番号で結果を保存 ビジュアル テストまたはスクリプトの再生の結果は、新しい増分実行番号で保存されます。たとえば、新規の結果は実行番号 1 で保存されます。同じビジュアル テストまたはスクリプトの後続の結果は、実行番号 2、3、4 など保存されます。これはデフォルトの設定です。

現在の実行番号に結果を追加 現在の再生の結果は、最新の実行番号の結果に付加されます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。


既存の実行番号を新しい結果で上書き 現在の再生の結果で、最新の既存の実行番号の結果が上書きされます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。

この結果を保存しない ビジュアル テストまたはスクリプトを再生したときに結果は作成されません。

5. 省略可能：**オプション** をクリックして、結果オプションを変更します。
詳細については、「再生結果オプションの設定」を参照してください。
6. 省略可能：**ローカル変数ウィンドウの表示** チェック ボックスをオンにすると、再生中に **ローカル変数** ウィンドウが表示されます。
Silk Test Workbench は、再生が終わると、**ローカル変数** ウィンドウを非表示にします。
7. **OK** をクリックします。
8. Web アプリケーションをテストする場合は、**ブラウザーの選択** ダイアログ ボックスが開きます。ブラウザーを選択して、**実行** をクリックします。
9. 省略可能：スクリプトを停止する場合は、**操作 > 停止** を選択するか、ツールバーの **停止** をクリックするか、両方の **Shift** キーを同時に押します。現在のステップで再生を一時停止する場合は、**操作 > 一時停止** を選択するか、ツールバーの **一時停止** をクリックするか、**Alt + F11** キーを押します。

 **注：**再生中にエラーが発生した場合は、エラー メッセージが表示されます。

- 10再生が完了したら、**再生完了** ダイアログ ボックスを使用して、実行する操作を決定します。

 **注:** 再生中にエラーが発生した場合は、エラーメッセージが表示されます。

再生中、**再生** ダイアログ ボックスを非表示にするには、**このダイアログを再表示しない** チェック ボックスをオンにします。これにより、全般オプションの **[再生] ダイアログの表示** オプション設定が上書きされます。**再生** ダイアログ ボックスを表示するには、この設定を使用します。

再生中のオートメーション操作の表示

出力 ウィンドウにオートメーション操作を表示して、スクリプトを確認またはトラブルシューティングします。

1. **表示** > **出力** を選択します。出力 ウィンドウが開きます。
2. **ツール** > **オプション** をクリックします。オプション ダイアログ ボックスが開きます。
3. オプション メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。再生 のサブカテゴリが表示されます。
4. **全般** をクリックします。
5. **出力ウィンドウにオートメーション操作を表示する** リスト ボックスから、**はい** を選択します。

スクリプトを再生して、関連するオートメーション操作を表示します。


テストの並列実行


複数の Workbench プロセスを使用して、複数のブラウザーやモバイル デバイスに対してテストを並列に実行することができます。たとえば、CI サーバーや Silk Central からテストを実行するときに、この機能を使用できます。

デフォルトでは、Silk Test は、次のブラウザーやプラットフォームでの並列テストをサポートします。

- Google Chrome
- Mozilla Firefox
- 以下のプラットフォーム上での Web、ネイティブ、ハイブリッド アプリ
 - 物理 Android デバイス
 - Android エミュレータ
 - 物理 デバイス

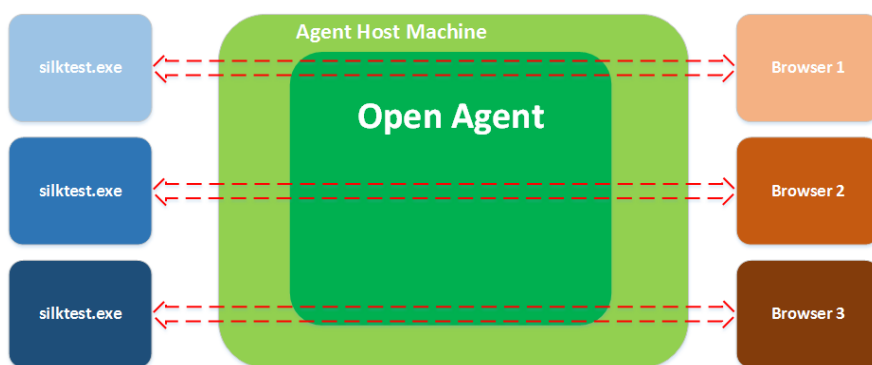
テストの並列再生を無効にするには、環境変数 `SILKTEST_ENABLE_PARALLEL_TESTING` を `false` に設定します。

 **注:** 並列テストを有効にすると、Open Agent はテストの実行プロセスのそれぞれを別々に処理しようとします。ある Silk Test クライアントでテストされたアプリケーションは、最初のクライアントが実行している間、他のクライアントでテストすることはできません。たとえば、Silk4J と Silk4NET で交互に同じアプリケーションをテストすることはできません。

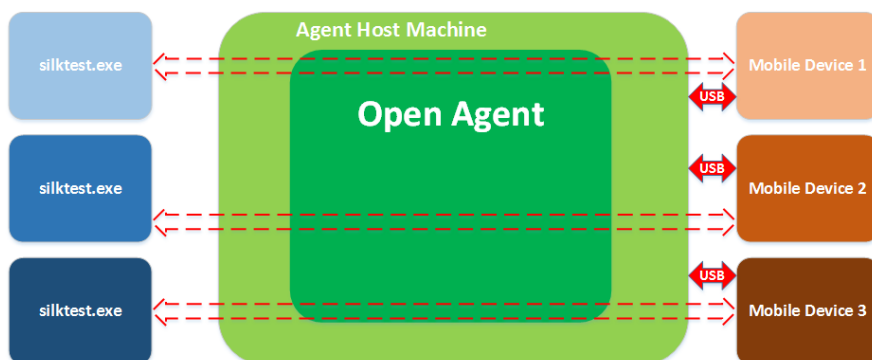
 **注:** 同じモバイル デバイス上で同時に複数のテストを実行することはできません。テストを並列実行する前に、十分なデバイスとエミュレータが利用できることを確認してください。モバイル デバイスやエミュレータが割り当てられないと、テストの実行は失敗します。

並列テストの実行それぞれが、1 個のブラウザーまたはモバイル デバイスに対して個々の `silktest.exe` を開始します。ブラウザーやモバイル デバイスを特定の `silktest.exe` と関連付けたい場合には、接続文字列を使用できます。詳細については、「[モバイル デバイスの接続文字列](#)」または「[リモートデスクトップ ブラウザーの接続文字列](#)」を参照してください。

次の図は、複数のブラウザーの並列テストのイメージを表します。



次の図は、複数のデバイスの並列テストのイメージを表します。



複数のプロセスが同時に開始されると、Silk Test Workbench を実行しているマシン上で Open Agent を何度も開始しようとする可能性があります。同じマシンで複数回 Open Agent を実行することはできないため、Silk Test Workbench は例外をスローすることになります。これを避けるため、テストを並列実行する前に、Open Agent が実行されていることを確認してください。

テスト結果は、複数の 結果ファイル（各テスト実行毎に 1 つ）に保存されます。

ビジュアル テスト再生の構成

ビジュアル テストで検証が処理される方法のようなビジュアル テストの再生動作、ビジュアル テストのテスト アプリケーションとのタイミングと同期、結果の使用方法和結果に表示される内容など、さまざまな側面を構成できます。

ビジュアル テストの再生動作を構成する方法は主に 2 つあります。最初の方法は、さまざまな再生オプションの値を **オプション** ダイアログ ボックスで設定します。**オプション** ダイアログ ボックスで設定する再生オプションは、Silk Test Workbench によってその後のすべてのビジュアル テストの再生に自動的に適用されるグローバル オプションです。これらの構成オプションを設定するには、**ツール > オプション** を選択します。

2 番めの方法は、ビジュアル テストに再生設定の設定ステップを挿入し、グローバル再生オプション値を上書きします。再生設定の設定ステップに再生オプションを設定しても、**オプション** ダイアログ ボックスに表示されるデフォルトのグローバル再生オプションは永続的に変更されるわけではありません。再生設定の設定ステップでは、ステップが表示されているビジュアル テストの再生オプションのみが変更されます。

また、再生オプションの現在の値をビジュアル テストの変数に返して、この変数を式やテスト ロジックなどで使用することができます。

ビジュアル テスト内での再生設定値の設定

ビジュアル テスト内で再生設定を設定して、**オプション** ダイアログ ボックスのグローバル再生オプションのデフォルト値を上書きすることができます。ビジュアル テスト内で再生オプションを上書きしても、**オプション** ダイアログ ボックスに表示される再生オプションのデフォルト値が永続的に変更されるわけではありません。

1. 再生設定の値を設定するビジュアル テストを開きます。
2. 再生設定ステップを挿入する場所の直前のステップを選択します。
3. **挿入 > 再生設定 > 設定ステップ** を選択します。Silk Test Workbench によって、選択したステップのあとに再生設定の設定 ステップが挿入されます。再生設定の設定 ステップのプロパティが**プロパティ** ペインに表示されます。
4. 再生設定の設定 ステップを選択して、**プロパティ** ペインにそのプロパティを表示します。
5. **プロパティ** ペインで、目的の再生設定を選択し、その値を割り当てます。

以下の設定を利用できます。

- **ツール > オプション > 詳細設定** 以下のすべてのオプション (**ツール > オプション > 詳細設定 > アクティブ データ** 以下のオプションを除く)。
- **ツール > オプション > 再生 > 全般** 以下のすべてのオプション。
- **ツール > オプション > 再生 > タイミング** 以下のすべてのオプション。
- **ツール > オプション > 再生 > xBrowser** 以下のすべてのオプション。
- **ツール > オプション > 再生 > 閉じる** 以下のすべてのオプション。
- **ツール > オプション > 再生 > 再生ステータス ダイアログ** 以下のすべてのオプション。

Silk Test Workbench は、再生設定の設定ステップのラベルを次のように更新します：

再生設定 '[再生設定名]' に '[値]' を設定します

6. 他の再生設定の設定 ステップを作成するには、必要に応じて上記の手順を繰り返します。ビジュアル テストを再生すると、再生設定の設定 ステップで指定した再生設定値が Silk Test Workbench によって使用されます。

ビジュアル テスト内での再生設定値の取得

オプションの現在値を返して、ローカル変数に格納するステップをビジュアル テストに挿入することにより、再生設定を取得することができます。このローカル変数は、検証やロジックのステップで使用したり、再生設定値を元の値にリセットするために使用したりできます。

1. 再生設定の値を取得するビジュアル テストを開きます。
2. 再生設定値を保存するローカル変数を作成します。
3. 再生設定ステップを挿入する場所の直前のステップを選択します。
設定値を取得して変数に設定するステップは、その変数を使用するステップの直前に実行する必要があります。
4. **挿入 > 再生設定 > 取得ステップ** を選択します。Silk Test Workbench によって、選択したステップのあとに再生設定の取得ステップが挿入されます。再生設定の取得ステップのプロパティが**プロパティ** ペインに表示されます。
5. 新しく作成されたステップを選択して、**プロパティ** ペインにそのプロパティを表示します。
6. 適切な設定を選択して、その値を変数に割り当てます。

以下の設定を利用できます。

- **ツール > オプション > 詳細設定** 以下のすべてのオプション (**ツール > オプション > 詳細設定 > アクティブ データ** 以下のオプションを除く)。
- **ツール > オプション > 再生 > 全般** 以下のすべてのオプション。

- ツール > オプション > 再生 > タイミング 以下のすべてのオプション。
- ツール > オプション > 再生 > xBrowser 以下のすべてのオプション。
- ツール > オプション > 再生 > 閉じる 以下のすべてのオプション。
- ツール > オプション > 再生 > 再生ステータス ダイアログ 以下のすべてのオプション。

Silk Test Workbench は、再生設定の取得ステップのラベルを次のように更新します：

再生設定 '[再生設定名]' を取得して、ローカル変数 '[変数名]' に格納します

7. 他の再生設定の取得ステップを作成するには、必要に応じて上記の手順を繰り返します。

挿入されたビジュアル テストまたは VB .NET スクリプトの再生の構成

ビジュアル テストに VB .NET スクリプトまたはビジュアル テストを挿入したあとは、挿入した VB .NET スクリプトまたはビジュアル テストに適用する再生設定を構成することができます。

1. 挿入された VB .NET スクリプトまたはビジュアル テストが含まれているビジュアル テストを開きます。

2. VB .NET スクリプトまたはビジュアル テストを実行するステップを選択します。

ビジュアル テストのステップ テキストは、次のようになります：

ビジュアル テスト [資産名] の再生

ここで、[資産名] は、挿入されたビジュアル テストの名前です。

3. プロパティ ペインを開きます。

4. 再生設定 リストから、挿入された VB .NET スクリプトまたはビジュアル テストの再生時に適用する設定を次の中から選択します。

- システムのデフォルトを使用
- 親の設定を継承する
- 親から設定をコピー

Silk Test Workbench がテストを同期する方法

ほとんどの予期しないテストの失敗は、同期問題に起因します。テストの再生時の同期が弱いと、予期しない結果が生成されるため、実際のアプリケーションの問題を見つけることが困難になります。同期エラーはタイミングの問題で、テストの環境に強く依存しており、このようなエラーを再現し解決することは困難です。たとえば、特定のテスト環境で発生する同期エラーが、開発環境では再現できないことがあります。同期が弱いと、自動化プロジェクトにおいて、自動化テスト セットの肥大化に伴い管理不能に陥る、最も一般的な理由のひとつになります。

Silk Test Workbench では、自動的なテスト同期をすべてのサポートするテクノロジーに対して提供しているため、ロバストで管理可能なテスト セットを構築できます。テストの再生中に、Silk Test Workbench は、AUT が次の操作を実行する準備ができるまで、常にテストが待機するようにします。テストの検証ステップでは、Silk Test Workbench は、テストのそれより前のテストが検証を実行する前に完了しているようにします。

AUT の特定の動作にテストを適用する場合は、次の同期タイムアウトの値を変更できます。

同期タイムアウト (OPT_SYNC_TIMEOUT)

再生中に AUT が準備完了状態になるまで Silk Test Workbench が待機する最大時間をミリ秒で指定します。デフォルト値は、300000 ミリ秒です。

オブジェクト解決タイムアウト (OBJ_WAIT_RESOLVE_OBJDEF)

Find メソッドがオブジェクトを検索する最大時間をミリ秒で指定します。デフォルト値は、5000 ミリ秒です。



注: 低速な接続でアクセスした仮想マシンなどの低速なシステムや高負荷な環境でテストを正しく実行できるようにするために、Silk Test Workbench は内部タイムアウト値を増やすことがあります (AUT の開始時やダイアログやウィンドウの表示時など)。AUT、ダイアログ、ウィンドウが完全に表示されると、Silk Test Workbench は再びタイムアウト値を OPT_WAIT_RESOLVE_OBJDEF で指定された値に戻します。

オブジェクト解決再試行間隔 (OPT_WAIT_RESOLVE_OBJDEF_RETRY)

Silk Test Workbench がオブジェクトを直ちに見つけることができなかった場合、Silk Test Workbench はオブジェクト解決タイムアウトが経過するまでオブジェクトの検索を再試行します。オブジェクト解決再試行間隔は、Silk Test Workbench がオブジェクトの検索を再試行するまで待機する時間をミリ秒で指定します。デフォルト値は、1000 ミリ秒です。

オブジェクト有効化タイムアウト (OPT_OBJECT_ENABLED_TIMEOUT)

再生中にオブジェクトが有効になるまで Silk Test Workbench が待機する最大時間をミリ秒で指定します。デフォルト値は、1000 ミリ秒です。



注: タイムアウトは重なりません。

Silk Test Workbench が提供する自動的な同期のうち、特に Web アプリケーションに関する詳細な情報については、「xBrowser のページ同期」を参照してください。Silk Test Workbench が提供する同期のうち、特に AJAX アプリケーションに関する詳細な情報については、「[How to Synchronize Test Automation Scripts for Ajax Applications](#)」を参照してください。

自動同期の他にも、Silk Test Workbench では、VB .NET スクリプトに待機関数を手動で追加することも可能です。手動同期を行うために、Silk Test Workbench で提供されてる待機関数は次の通りです。

WaitForObject	指定したロケータに一致するオブジェクトを待機します。XPath ロケータやオブジェクト マップ識別子に対して動作します。
WaitForProperty	指定したプロパティが指定した値になるか、タイムアウト値に到達するまで待機します。
WaitForPropertyNotEquals	指定したプロパティの値が指定した値でなくなるか、タイムアウト値に到達するまで待機します。
WaitForDisappearance	オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。
WaitForChildDisappearance	locator パラメータで指定された子オブジェクトが存在しなくなるか、タイムアウト値に到達するまで待機します。
WaitForScreenshotStable	オブジェクトの表示が安定し、その位置が変化しなくなるまで待機します。

テストが ObjectNotFoundException で不規則に失敗する場合、オブジェクト解決タイムアウトを増やします (30 秒にするなど)。進行状況ダイアログが表示され、長時間計算が行われた後に表示されるオブジェクトをクリックする場合など、特定の操作に特別に時間のかかる場合は、WaitForObject メソッドをテス

ト スクリプトに手動で追加してオブジェクトが見つかるまで待機するようしたり、WaitForDisappearance メソッドをテスト スクリプトに追加して進行状況ダイアログが表示されなくなるまで待機するようになります。

自動的な同期の例

Silk Test Workbench がキャプション **Ok** を持つボタンをクリック する、次のサンプルコードを考えます。

```
'VB .NET code
Dim button = _desktop.PushButton("@caption='Ok'")
button.Click()
```

このサンプル コードの操作を再生するために、Silk Test Workbench は次の同期操作を実行します。

1. Silk Test Workbench はボタンを検索します。そして、オブジェクト解決タイムアウトが経過した場合、Silk Test Workbench は再生を停止して例外をスローします。
2. Silk Test Workbench はテスト対象アプリケーション (AUT) が準備完了状態になるまで待機します。AUT が準備完了状態になる前に、同期タイムアウトが経過した場合、Silk Test Workbench は再生を停止して例外をスローします。
3. Silk Test Workbench はボタンが有効になるまで待機します。オブジェクト有効化タイムアウトがボタンが有効になる前に経過した場合、Silk Test Workbench は再生を停止して例外をスローします。
4. Silk Test Workbench はボタンをクリックします。
5. Silk Test Workbench はテスト対象アプリケーション (AUT) が再び準備完了状態になるまで待機します。

結果の分析

結果 はビジュアル テストやスクリプトの再生中に生成される資産で、これによって、テスト プロセスを証明したり、任意の時点におけるテスト アプリケーションの機能の状態を永続的に記録したりできます。結果には、ビジュアル テストやスクリプトの名前、実行番号、各ステップや行が実行された日時、コマンドやステップのステータス (合格か失敗か)、その他の重要な情報など、再生に関する情報が含まれます。結果は、再生情報を確認し、テストに失敗した領域を分析するのに役立ちます。

結果の表示は、再生後すぐに **結果** ウィンドウを開くか、または **アセット ブラウザ** から結果を選択して行います。**結果** ウィンドウでは、**ビジュアル ナビゲータ** の 4 つのペインを使用して結果情報が表示されます。また、**結果** ウィンドウでは結果情報がタブを使用して整理されるため、再生の要約レポート、合格/失敗の検証、フラグを付けられたステップ、およびビジュアル テストのすべてのステップやスクリプトのコード行の詳細な結果にすばやくアクセスできます。

結果の **画面プレビュー** で、ビジュアル テストの記録中にキャプチャされていた画面と再生中にキャプチャされた画面とを比較できます。比較して違いがあった場合は、**画面プレビュー** からすぐに再生画面のビジュアル テスト画面を更新できます。詳細については、「画面テスト ステップのキャプチャ画面の更新」を参照してください。

テスト手順の文書化や、他のテスト担当者との協業においては、結果を印刷したり、結果をファイルに保存したり、結果内の重要な問題について自分自身または他の担当者に注意を促すために、項目にフラグを付けたりすることができます。


また、フィルタを作成して適用すれば、最も関心のある結果情報のみを表示できます。たとえば、直前のテストの再生で失敗した検証のみを確認する場合、フィルタを作成して適用し、必要な情報のみを表示することができます。


結果を開いたあと、Silk Test Workbench では **結果** ウィンドウが表示されます。このウィンドウで、ビジュアル テストまたはスクリプトの再生に関するすべての側面についての詳細な情報を確認および分析できます。

[結果] ウィンドウ

ビジュアル テストまたはスクリプトを再生したあと、再生結果を **結果** ウィンドウに表示することができます。

Summary **Passed** Failed Flags Details (16 steps)

Overall result for run 1 :  **Passed** No verifications were executed

Browser:  Internet Explorer 11 - Windows 7

Reason for abort: Not applicable

Latest run number: 1

Recent runs: 1

Visual test executed: TestResultOS

Visual test description:

Result description:

Scripts (number of times each ran): [Project A.TestResultOS.version 1 \(1\)](#)

Verifications passed: 0 / 0

Verifications failed: 0 / 0

Flags: 0

Start time: 13.03.2018 14:14:39

End time: 13.03.2018 14:14:46

Total time: 7 s

Steps run: 16

Playback setting: System Defaults Edit...

結果 ウィンドウには以下の機能があります。

ビジュアル ナビゲータ テスト再生の全体の様子がすばやくわかります。詳細については、「[結果] ウィンドウのビジュアル ナビゲータ」を参照してください。

[結果] ウィンドウの ツールバー 結果の内容の表示と種類をカスタマイズすることができます。詳細については、「[結果] ウィンドウのツールバー」を参照してください。

[結果] ウィンドウの タブ タブには、結果の内容を特定の種類にまとめられています。詳細については、「[結果] ウィンドウのタブ」を参照してください。

[結果] ウィンドウのツールバー

結果 ウィンドウのツールバーには、結果に含まれる内容の表示と種類をカスタマイズするための以下のボタンとドロップダウン リストがあります。

- **種類の選択によるフィルタ処理の結果**：すべての事前定義済みおよびユーザー定義の結果フィルタにすばやくアクセスできます。リストからフィルタを選択します。
- **フィルタの管理**：結果フィルタを作成、編集、複製、および適用できる **フィルタの管理** ダイアログ ボックスが開きます。
- **条件**：条件 ダイアログ ボックスが表示されます。このダイアログ ボックスでは、今後のすべての実行の成功を定義する条件として、合格した検証のパーセンテージを設定できます。たとえば、合格条件を 90% に設定した場合は、ビジュアル テストまたはスクリプトにおける 10 件の検証のうち 9 件以上に

合格しないと、再生の結果が合格になりません。このオプションを設定すると、**結果合格条件（パーセンテージ）** オプションが更新されます。このパーセンテージは、今後のすべての結果に適用されます。

- **すべての実行を表示：実行の詳細** ダイアログ ボックスが開き、テストのすべてのテスト実行結果、およびテスト実行が合格したか失敗したかの一覧が表示されます。このダイアログ ボックスで、結果を開いたり削除したりすることができます。
- **更新**：現在の結果を更新します。
- **表示：テスト ステップ** ペインに表示されるステップのタイプを設定します。このボタンの横にあるドロップダウン矢印をクリックして、ステップのみを表示するか、画面のみを表示するか、またはその両方を表示するかを選択します。**ステップの説明** 列を表示するように選択することもできます。選択された表示は、**結果** ウィンドウの各タブ（**合格**、**失敗**、**フラグ**、および **詳細**）に適用されます。
- **基本表示：テスト ステップ** ペインの標準の列が表示され、**プロパティ** ペインに対応するプロパティが表示されます。
- **詳細表示：テスト ステップ** ペインに追加の列、および **プロパティ** ペインに対応するプロパティが表示されます。**テスト ステップ** ペインに追加の列を表示するには、右にスクロールすることが必要となる場合があります。

[結果] ウィンドウのタブ

結果 ウィンドウのタブは、ビジュアル テストの特定の種類のステップまたはスクリプトの特定の種類のコード行をすばやく表示できるフィルタとして機能します。

合格、**失敗**、**フラグ**、および **詳細** タブのすべてで同じ方法、つまり **ビジュアル ナビゲータ** のペインを使用して内容が整理され、表示されます。それぞれのタブでは、表示される内容の種類が異なります。**詳細** タブには、各ステップまたはコード行が含まれます。**合格** タブには、合格した検証のみが含まれます。**失敗** タブには、失敗した検証のみが含まれます。**フラグ** タブには、フラグを付けられた結果ステップのみが含まれます。

結果 ウィンドウには以下のタブがあります。

要約 タブ 結果を開くと、このタブがデフォルトで表示されます。デフォルトの結果表示を変更するには、[全般] オプションの **デフォルトの結果表示** オプションを変更します。**要約** タブには、以下の情報を含む一般的な概要レポートが表示されます。

- **総実行結果**：ビジュアル テストまたはスクリプトが正常に再生されて、結果合格条件（パーセンテージ）を満たしている場合は、[合格] と表示されます。結果合格条件（パーセンテージ）を満たしていない場合は、[失敗] と表示されます。ステップを正しく実行できなかった場合は、[再生エラー] と表示されます。
- **ブラウザー**：Web アプリケーションに対してテストが実行された場合、次の情報が行に表示されます。
 - テストが実行されたブラウザー。
 - ブラウザーのバージョン。
 - 再生時に使用するブラウザー サイズ。この情報は、デフォルト サイズではなく、ブラウザー サイズを変更した場合にのみ表示されます。
 - オペレーティング システムとバージョン。
 - モバイル デバイス、エミュレータ、シミュレータの名前。モバイル Web アプリケーションに対してテストが実行された場合にのみ、この情報が表示されます。
 - ブラウザー テストが実行されたモバイル デバイスが接続されている、またはエミュレータやシミュレータが実行されているマシンの名前。モバイル Web アプリケーションに対してテストが実行された場合にのみ、この情報が表示されます。
- **デバイス**：ネイティブ モバイル アプリケーションに対してテストが実行された場合、次の情報が行に表示されます。
 - モバイル デバイス、エミュレータ、シミュレータの名前。
 - モバイル デバイス、エミュレータ、シミュレータのオペレーティング システムとそのバージョン。

- モバイル デバイスが接続されている、またはエミュレータやシミュレータが実行されているマシンの名前。
- **モバイル アプリケーション**：テストを実行したモバイル アプリケーションへの完全パス。ネイティブ モバイル アプリケーションに対してテストが実行された場合にのみ、この情報が表示されます。
- **中止の理由**：ビジュアル テストまたはスクリプトの再生が中止された理由を表示します。
- **最新の実行番号**：最新の結果の実行番号を表示します。
- **最近の実行**：最近の実行を表示します。前の実行を表示するには、表示する実行をクリックします。このフィールドに表示されていない以前の実行を開くには、ツールバーの **すべての実行を表示** をクリックして **実行の詳細** ダイアログ ボックスを開きます。ここで、前の実行を開いたり削除したりできます。
- **実行されたビジュアル テスト/.NET スクリプト**：結果のビジュアル テストまたはスクリプトの名前を表示します。
- **ビジュアル テスト/.NET スクリプトの説明**：結果のビジュアル テストまたはスクリプトの説明を表示します。
- **結果の説明**：結果の説明を表示します。
- **ビジュアル テスト** または **スクリプト**：再生の一部として正しく実行されたビジュアル テストまたはスクリプトをすべて一覧します。これには、Workbench.RunScript() メソッドを使用して実行された、挿入されたビジュアル テストまたはスクリプトも含まれます。たとえば、ドライバ スクリプトでは、一度の再生により複数のスクリプトが実行されることがあります。さらに、ビジュアル テストまたはスクリプトのバージョンと、ビジュアル テストまたはスクリプトを実行した回数が表示されます。
- **合格した検証**：すべてのビジュアル テストまたはスクリプトで、正常に実行され合格した検証の合計数を表示します。このフィールドの数字をクリックすると **合格** タブが開くため、合格したすべての検証を確認できます。
- **失敗した検証**：すべてのビジュアル テストまたはスクリプトで、正常に実行されたが失敗した検証の合計数を表示します。このフィールドの数字をクリックすると **失敗** タブが開くため、失敗したすべての検証を確認できます。
- **フラグ**：ビジュアル テストについて、結果でフラグを付けられた検証ステップの総数です。このフィールドの数値をクリックして、**フラグ** タブを開きます。フラグは、スクリプトでは使用できません。
- **開始時刻**：最初のビジュアル テストまたはスクリプトの再生が開始された時刻です。
- **終了時刻**：最後のビジュアル テストまたはスクリプトの再生が終了した時刻です。
- **合計時間**：ビジュアル テストまたはスクリプトを再生した合計時間です。
- **実行したステップ数**：実行したステップまたはコード行の総数です。
- **再生設定**：結果の作成に使用される再生オプションのグループである再生設定を表示します。**編集** をクリックして再生オプションを表示し、再生設定を設定することができます。

合格 タブ： 再生中に合格した検証が表示されます。

失敗 タブ： 再生中に失敗した検証が表示されます。再生エラーとなったステップはこのタブには表示されません。

フラグ タブ： 結果内のフラグ付きステップが表示されます。結果が作成されたあとで手動でフラグを挿入するか、またはビジュアル テストの再生中にフラグを挿入する検証ロジックを使用することで、結果のステップにフラグを挿入できます。**フラグ** タブには、結果内のすべてのフラグ付きステップが動的に表示されます。たとえば、**合格**、**失敗**、または **詳細** タブのいずれかにフラグを挿入すると、Silk Test Workbench によって **フラグ** タブが自動的に更新され、フラグを付けられたステップが表示されます。



注： ビジュアル テストのフラグ付きのステップは、再生後はこのタブに表示されません。結果でフラグを付けられたステップと検証ロジックで挿入されたフラグのみが、結果の **フラグ** タブに表示されます。フラグは、スクリプトでは使用できません。

詳細タブ: ビジュアル テストの各ステップまたはスクリプトの各コード行に関する情報が表示されます。テスト ステップやコード行の名前、合格/失敗のステータス、説明、フラグのステータスなどです。キーワード駆動テストの場合、**詳細** タブにはキーワード駆動テストに含まれ、ビジュアル テストとして実装されているキーワードのステップが表示されます。**詳細** タブには、再生の詳細も表示されます。再生中にエラーが発生したとき、そのビジュアル テストまたはスクリプトが他のビジュアル テストから呼び出されていた場合、再生の詳細には、ビジュアル テストまたはスクリプトを呼び出したビジュアル テストのテスト ステップも含まれます。

[結果] ウィンドウのそれぞれのタブで、**テスト ステップ** ペインの **基本表示** または **詳細表示** のいずれかを選択できます。**詳細表示** では、**テスト ステップ** ペインに追加の列が表示されます。


実行ステータスの種類

Silk Test Workbench は、再生中のテストのパフォーマンスに基づいて、結果の各テスト実行に対して実行ステータスを割り当てます。実行ステータスは、結果の **要約** タブの **総実行結果** テキスト ボックスに表示されます。

実行ステータスは結果の合格条件を示す割合に依存します。結果の合格条件は、テストの合格を定義する、合格した検証の割合を指定したものです。たとえば、合格条件を 90% に設定した場合、テストの 10 件の検証のうち 9 件に合格しないと、再生の結果が合格になりません。

実行ステータスの種類

実行ステータスの種類には、以下のものがあります。

タイプ	説明
合格	合格 実行ステータスは、成功したテスト再生を示します。テストに検証が含まれている場合、 合格 ステータスは結果の合格条件の割合に達していることも示します。
失敗	失敗 実行ステータスは、再生中に失敗した検証がテストに含まれており、失敗した検証の割合が結果の合格条件の割合に達していないことを示します。  注: デフォルトでは、結果の合格条件は 100% に設定されています。このオプションを変更するには、 結果 ウィンドウのツールバーの 条件 をクリックします。
再生エラー	再生エラー 実行ステータスは、正常に実行されなかったステップが結果に含まれていることを示します。たとえば以下のような場合です。 <ul style="list-style-type: none"> アクティブ データ ファイル内の存在しない列を参照するステップ 存在しないローカル変数を参照するステップ 開始されていないタイマーを実行するステップ

結果のステップのステータスの詳細については、「ステップのステータスの種類」を参照してください。


ステップのステータスの種類

Silk Test Workbench は、再生中のステップのパフォーマンスに基づいて、結果の各ステップに対してステータスを割り当てます。ステップのステータスは **テスト ステップ** ペインの **結果** 列に表示され、ステップのテキストはそのステータスの種類に応じた色で表示されます。

ステップのステータスの種類

結果のステップのステータスの種類には、以下のものがあります。

タイプ	説明
合格	検証ステップがそのステップの検証ロジックで定義されている条件を満たしている場合は、ステップに 合格 ステータスが割り当てられます。

タイプ	説明
失敗	<p>このステータスの種類が割り当てられているステップは、結果の 合格 タブおよび 詳細 タブに表示されます。 結果 列に 合格 と表示され、ステップ テキストが緑で表示されます。</p> <p>検証ステップがそのステップの検証ロジックで定義されている条件を満たしていない場合は、ステップに 失敗 ステータスが割り当てられます。</p> <p>このステータスの種類が割り当てられているステップは、結果の 失敗 タブおよび 詳細 タブに表示されます。 結果 列に 失敗 と表示され、ステップ テキストが赤で表示されます。</p>
再生エラー	<p>再生エラー ステータスのステップは、正常に実行されなかったステップです。たとえば以下のような場合です。</p> <ul style="list-style-type: none"> • アクティブ データ ファイル内の存在しない列を参照するステップ • 存在しないローカル変数を参照するステップ • 開始されていないタイマーを実行するステップ <p>結果 列に 再生エラー と表示され、ステップ テキストが青で表示されます。</p>
エラー画面	<p>再生エラー ステータスのステップのあとに表示されるステップです。 エラー画面 ステータスが割り当てられたステップでは、テストで再生エラーが発生したときに表示されたテスト アプリケーションの画面が表示されます。</p> <p>結果 列に エラー画面 と表示され、ステップ テキストが青で表示されます。</p> <p> 注: このステップ タイプはテスト スクリプトでは利用できません。</p>
正しく実行されたステップ	<p>正しく実行されたすべてのステップ（検証ステップ以外）。</p> <p>結果 列は空で、ステップ テキストの色は黒です。</p>

結果の実行ステータスの詳細については、「実行ステータスの種類」を参照してください。

[結果] ウィンドウのビジュアル ナビゲータ

結果 ウィンドウには [ビジュアル ナビゲータ] があり、テストの再生をあらゆる面からすばやく確認できます。

ビジュアル テストの場合は、**合格** タブ、**失敗** タブ、**フラグ** タブ、**詳細** タブのすべてに [ビジュアル ナビゲータ] の以下の 4 つのペインが表示されます。

- テスト ステップ
- 画面プレビュー
- プロパティ
- ストーリーボード

結果 ウィンドウの **合格**、**失敗**、**フラグ**、および **詳細** タブでは、ビジュアル ナビゲータの 4 つのペインで内容が整理されて表示されます。**テスト ステップ**、**画面プレビュー**、**ストーリーボード**、および **プロパティ** ペインは互いに同期し、**テスト ステップ** ペインで選択したステップに固有の情報が表示されます。**結果** ウィンドウのビジュアル ナビゲータ ペインには、以下のセクションで説明する追加機能が含まれていません。

テスト ステップ

結果を表示すると、**テスト ステップ** ペインには、ビジュアル テストと同様の情報と各テスト ステップの結果の追加データが表示されます。ステップを選択すると、選択したステップに固有の情報でその他のペインが更新されます。また、フラグを挿入したり、選択したステップにステップの説明を追加したりできます。結果を [基本表示] または [詳細表示] で表示することを選択できます。

画面プレビュー

結果の [画面プレビュー] で、ビジュアル テストの記録中にキャプチャされていた画面と再生中にキャプチャされた画面とを比較できます。**テスト ステップ** ペインで比較する画面を含むステップを選択してから、**画面プレビュー** で **操作 > 相違点の表示 > 横に並べる** の順にクリックします。再生中にキャプチャされた画面を含むようにビジュアル テストを更新する場合は、**操作 > 画面の更新** の順にクリックします。

ストーリーボード

結果の **ストーリーボード** ペインでは、アイコンで検証ロジック ステップの実行ステータスが示されます。

プロパティ

結果の **プロパティ** ペインには、再生中にキャプチャされた結果のプロパティと値が表示されます。

プロパティ ペインには **再生前のビジュアル テストのステップ プロパティを表示/非表示** ツールバー ボタンがあり、選択したステップのビジュアル テストのプロパティを表示または非表示にすることができます。

スクリプトでは、**結果** ウィンドウには **テスト ステップ** ペインおよび **プロパティ** ペインのみ表示されません。**テスト ステップ** ペインには、各テスト ステップの再生ステータスと結果の詳細情報を示す列があります。

結果データの表示をカスタマイズするには、**全般** オプションから **デフォルトの結果表示** を変更します。再生結果オプションを変更し、目的のデータを表示するフィルタを作成することもできます。

結果 ウィンドウのツールバーの [基本表示] または [詳細表示] オプションを使用することにより、**プロパティ** ペインに特定のプロパティを表示できます。

結果の管理

デフォルトでは、Silk Test Workbench によりビジュアル テストまたはスクリプトの再生後に結果が作成され、固有の資産として保存されます。結果を表示するには、以下のいずれかの方法を使用できます。

- 再生後、**再生完了** ダイアログ ボックスから、手動で結果を開きます。
- 再生直後、自動的に結果を開く
- あとでアセット ブラウザから結果を開く

結果を開くと、**結果** ウィンドウが表示され、デフォルトで結果の **要約** タブが表示されます。デフォルトの結果表示を変更するには、**全般** オプションの **デフォルトの結果表示** オプションを変更します。

デフォルトの結果名は、再生したビジュアル テストまたはテスト スクリプトの名前です。**再生結果** オプションで **デフォルトの結果名** オプションを変更できます。

結果の作成

1. ビジュアル テストまたはスクリプトを開きます。
2. **再生** をクリックします。**再生** ダイアログ ボックスが開きます。
3. **結果の説明** テキスト ボックスに結果の説明を入力します。
4. **結果オプション** リストからオプションを選択することにより、結果を保存する方法を選択します。

以下のオプションのいずれかを選択します。

- **新しい実行番号で結果を保存** : 再生の結果は、増分された実行番号で保存されます。これはデフォルトの設定です。
- **現在の実行番号に結果を追加** : 現在の再生の結果は、最新の既存の実行番号の結果に追加されます。新しい実行番号は作成されません。この設定は、結果が存在している場合のみ表示されます。
- **既存の実行番号を新しい結果で上書き** : 前の再生は、新しい再生結果によって前の再生の実行番号を使って上書きされます。この設定は、ビジュアル テストまたはスクリプトがはじめて再生されたときにのみ表示されます。

- **この結果を保存しない**：結果ファイルを作成せずに再生します。既存の結果がないビジュアルテストまたはスクリプトに対して **現在の実行番号に結果を追加** または **既存の実行番号を新しい結果で上書き** が選択された場合は、**新しい実行番号で結果を保存** が使用されます。

5. **OK** をクリックします。

結果の動作と表示のカスタマイズ

Silk Test Workbench には、結果の動作と表示を設定するためのさまざまな方法があります。

オプション ダイアログ ボックスでは、結果オプションを含むさまざまな再生オプションの値を設定できます。**オプション** ダイアログ ボックスで設定する再生オプションは、Silk Test Workbench によってその後のすべてのビジュアルテストの再生に自動的に適用されるグローバル オプションです。詳細については、「再生オプションの変更」を参照してください。

ビジュアルテストの場合は、再生設定の設定ステップを挿入し、目的の再生設定を選択することにより、デフォルトの結果オプションを上書きできます。詳細については、「ビジュアルテスト内での再生設定値の設定」を参照してください。

スクリプトとビジュアルテストの場合は、結果の記録に関するいくつかの特性を **再生** ダイアログ ボックスから直接制御できます。詳細については、「結果の記録のオン/オフの切り替え」を参照してください。

結果の記録のオン/オフの切り替え

デフォルトでは、Silk Test Workbench は、ビジュアルテストまたはスクリプトの再生ごとに結果を作成します。再生後に Silk Test Workbench が結果を作成しないようにするには、**再生結果** オプションまたは **再生** ダイアログ ボックス（ビジュアルテストまたはスクリプトを再生するとデフォルトで表示されます）で **結果オプション** を設定します。

1. ツール > **オプション** を選択します。 **オプション** ダイアログ ボックスが開き、デフォルトで **全般** オプションがリストされます。
2. ツリー ビューで **再生** オプションをダブルクリックし、**結果** をクリックします。使用可能な結果オプションがダイアログ ボックスの右側のウィンドウに表示されます。
3. 結果の記録をオフにするには、**結果オプション** リストから **この結果を保存しない** を選択します。
4. 結果の記録をオンにするには、**結果オプション** リストから以下のいずれかのオプションを選択します。
 - **新しい実行番号で結果を保存**：再生の結果は、増分された実行番号で保存されます。これはデフォルトの設定です。
 - **現在の実行番号に結果を追加**：現在の再生の結果は、最新の既存の実行番号の結果に追加されます。新しい実行番号は作成されません。この設定は、結果が存在している場合にのみ表示されます。
 - **既存の実行番号を新しい結果で上書き**：前の再生は、新しい再生結果によって前の再生の実行番号を使って上書きされます。この設定は、ビジュアルテストまたはスクリプトがはじめて再生されたときにのみ表示されます。

5. **OK** をクリックします。

再生後に結果を手動で開く

再生後に手動で結果を開いて、結果を表示することができます。

1. ツール > **オプション** を選択します。 **オプション** ダイアログ ボックスが開き、デフォルトで **全般** オプションがリストされます。
2. **[再生完了] ダイアログの表示** オプションが **はい** になっていることを確認するか、または設定します。
3. ビジュアルテストまたはスクリプトを実行します。 **再生完了** ダイアログ ボックスが開きます。
4. **結果へ移動** をクリックします。 **結果** ウィンドウが開き、結果が表示されます。

再生後に結果を自動的に開く

自動的に結果を開くように Silk Test Workbench を設定して、結果を表示することができます。

1. ツール > オプション を選択します。オプション ダイアログ ボックスが開き、デフォルトで 全般 オプションがリストされます。
2. 結果を自動的に開く リスト ボックスから、はい を選択します。
3. [再生完了] ダイアログの表示 リスト ボックスから、いいえ を選択します。
これにより、再生後に 再生完了 ダイアログ ボックスが表示されなくなります。
4. ビジュアル テストまたはスクリプトを実行します。再生後、Silk Test Workbench によって自動的に 結果 ウィンドウが表示され、結果が表示されます。

アセット ブラウザからの結果の表示

あとでアセット ブラウザから結果を開き、結果を表示することができます。

1. 表示 > アセット ブラウザ を選択します。アセット ブラウザ ウィンドウが開きます。
2. 資産の種類 リストから、結果 を選択します。項目ペインに既存の結果がリストされます。
3. 表示する結果をダブルクリックします。結果 ウィンドウが開き、結果が表示されます。

複数のビジュアル テストのエラーのトレース

ビジュアル テストまたは VB .NET スクリプトが他のビジュアル テストから呼び出され、実行中にエラーが発生した場合、呼び出したビジュアル テストの結果の 詳細 タブにエラーを起こしたテストステップが表示され、エラーが発生した呼び出されたビジュアル テストまたは VB .NET スクリプトそれぞれのテストステップまたはコードの行数と共に表示されます。呼び出したビジュアル テストが、さらに他のビジュアル テストのステップから呼び出されていた場合、呼び出したビジュアル テストの結果には呼び出されビジュアル テストのステップがさらに表示されます。これらのテスト ステップを参照するには、つぎの作業を行います。

1. 実行されたビジュアル テストの結果を開きます。
2. 詳細 タブを選択します。
3. エラーが発生したステップを選択します。
4. プロパティ ペインを開きます。再生の詳細 セクションの 呼び出し元 リストに、呼び出したビジュアル テスト、および呼び出されたビジュアル テストまたはテスト スクリプトにおける、エラーが発生したステップが対応するビジュアル テストまたはテスト スクリプトの名前と共にリスト表示されます。
5. リストから資産を選択し、資産を開く をクリックすると、エラーが発生したステップまたは行に直接アクセスできます。

例

ビジュアル テスト *Test1* の 2 番目のステップがビジュアル テスト *Test2* を呼び出し、*Test2* の 5 番目のステップがビジュアル テスト *Test3* を呼び出している場合に、*Test1* の実行時に *Test3* の 4 番目のステップでエラーが発生すると、*Test1* の結果ファイルには次の項目が含まれます。

呼び出し元	
Test1	2
Test2	5
Test3	4

結果からビジュアル テストまたはテスト スクリプトへの切り替え

特定の結果にリンクされているビジュアル テストを表示します。

1. ビジュアル テストの結果を開きます。

2. **詳細** タブをクリックし、**テスト ステップ** ペインのタイトルバーで **操作** をクリックします。
3. **ビジュアル テストのウィンドウ** を選択します。
4. **結果** ウィンドウに戻るには、**ビジュアル テストの テスト ステップ** ペインのタイトルバーで **操作** をクリックして、**[結果] ウィンドウへ移動** を選択します。

ビジュアル テストへの結果コメントの挿入

結果コメントを使用すると、結果内のビジュアル テストに関する補足情報を提供できます。ビジュアル テストを再生すると、**結果** ウィンドウの **詳細** タブにコメントがステップとして表示されます。



ヒント: 結果コメントの挿入以外に、フラグを使用して選択したステップのコメントを挿入することもできます。

1. ビジュアル テストの場合、ビジュアル テストの **テスト ステップ** ペインで、結果コメントを表示するステップを選択します。
2. **挿入 > 結果コメント** を選択します。 **テスト ステップ** ペインに 結果コメント ステップが表示され、このステップのプロパティが **プロパティ** ペインに表示されます。
3. **プロパティ** ペインで、以下のステップのいずれかを実行します。
 - **式** プロパティ値テキスト ボックスにコメントを入力します。
 - テキスト ボックスをクリックして、**選択** ボタンを有効にすると、変数を選択したり、コメント値として使用する式を作成できます。
4. ビジュアル テストを再生します。結果の **詳細** タブで、結果コメント ステップの **結果の詳細** 列にコメントが表示されます。

ビジュアル テストの結果の印刷

1. 印刷する結果を開きます。
2. **ファイル > 印刷** を選択します。**印刷** ダイアログ ボックスが開きます。
3. リストからプリンタを選択します。
4. 省略可能：**プロパティ** をクリックすると、余白や印刷の向きなどの印刷ページ設定を変更できます。
5. 省略可能：**印刷範囲** セクションで、ドキュメント全体を印刷するか、または選択したページのみを印刷するかを指定します。
6. 省略可能：**部数** フィールドで、印刷する部数を指定します。
7. 省略可能：**部単位で印刷する** チェック ボックスをオンにすると、すべてのコピーの対応するページをまとめて印刷します。
このチェック ボックスは、複数の部数を印刷するように選択した場合にのみ有効になります。
8. 印刷出力をカスタマイズするには、**オプション** をクリックして、印刷出力に含める結果情報の種類を選択します。

以下のオプションがあります。

**以下を
含める** 含める要約の種類を指定します。

- **ビジュアル テストの要約**：要約にはビジュアル テストの名前、プロジェクト、作成者、作成日、バージョン、および説明が含まれています。この要約は、印刷出力の最後のページに表示されます。
- **結果の要約**：最新の結果の要約ページを含めます。この要約は、印刷出力の最初のページに表示されます。
- **グループ画面**：現在のダイアログ ボックスまたは Web ページのメイン コンテンツを含むグループ画面のスナップショットを印刷出力に含めます。グループ画面に続くテストステップは、現在のダイアログ ボックスまたは Web ページのコンテンツで発生する単一の操作に対応します。グループ画面は、**テスト ステップ** ペインの Using ステップに対応

します。グループ画面のスナップショットは、**画面プレビュー** ペインで表示されるように、印刷出力に示されます。

タブ 印刷対象の内容を含むタブを指定します。印刷出力では、**詳細** タブの内容が先に示され、そのあとに **合格** タブ、**失敗** タブ、および **フラグ** タブの内容が示されます。

- **合格** : 結果の **合格** タブの内容を含めます。
- **失敗** : 結果の **失敗** タブの内容を含めます。
- **フラグ** : 結果の **フラグ** タブの内容を含めます。
- **詳細** : 結果の **詳細** タブの内容を含めます。

各ステップに含める 各ステップに含める内容を指定します。

- **画面** : 各ステップの関連画面スナップショットを含めます。個々の画面は (ステップ画面) に対応し、現在のダイアログ ボックスまたは Web ページでの特定の操作に関連付けられます。ステップ画面のスナップショットは、**画面プレビュー** ペインで表示されるように、印刷出力に示されます。
- **説明** : 各ステップの説明を含めます。ステップの説明は、**プロパティ** ペインの **ステップの説明** プロパティに表示されるように、印刷出力に示されます。
- **フラグの詳細** : フラグの説明、作成者、更新者、更新日、割り当てられたユーザーなどのフラグの詳細を含めます。

以下のプロパティを含める 各ステップに含めるプロパティを指定します。

- **拡張プロパティ** : ステップに関する補足情報を提供する拡張プロパティを含めます。
- **その他すべてのプロパティ** : 各ステップの拡張プロパティ以外のすべてのプロパティを含めます。

デフォルト デフォルトの印刷オプションを選択するには、このボタンをクリックします。

9. 省略可能 : **プレビュー** をクリックして、結果情報がどのように印刷されるかを表示します。**プレビュー** ウィンドウが開きます。

100K をクリックします。

スクリプトの結果の印刷

1. 印刷する結果を開きます。
2. **ファイル** > **印刷** を選択します。**印刷** ダイアログ ボックスが開きます。
3. 印刷オプションを設定します。
4. 印刷出力をカスタマイズするには、**オプション** をクリックして、印刷出力に含める結果情報の種類を選択します。

以下のオプションがあります。

以下を含める 含める内容の種類を指定します。

- **.NET スクリプトの要約** : スクリプトの名前、プロジェクト、作成者、作成日、バージョン、および説明を含むスクリプトの要約を含めます。この要約は、印刷出力の最後のページに表示されます。
- **結果の要約** : スクリプトの最新の結果の要約ページを含めます。この要約は、印刷出力の最初のページに表示されます。
- **グループ画面** : スクリプトでは利用できません。

タブ 印刷対象の内容を含むタブを指定します。印刷出力では、**詳細** タブの内容が先に示され、そのあとに **合格** タブ、**失敗** タブ、および **フラグ** タブの内容が示されます。

- **合格** : 結果の **合格** タブの内容を含めます。

- **失敗** : 結果の **失敗** タブの内容を含めます。
- **フラグ** : 結果の **フラグ** タブの内容を含めます。
- **詳細** : 結果の **詳細** タブの内容を含めます。

各ステップに含める 各ステップに含める内容を指定します。

- **画面** : 各ステップに関連したスクリーンショットを含めます。個々の画面は、テスト対象アプリケーションのコントロールに対する固有の操作に関連付けられています。
- **説明** : スクリプトでは利用できません。
- **フラグの詳細** : フラグの説明、作成者、更新者、更新日、割り当てられたユーザーなどのフラグの詳細を含めます。

以下のプロパティを含める 各ステップに含めるプロパティを指定します。


- **拡張プロパティ** : スクリプトでは利用できません。
- **その他すべてのプロパティ** : スクリプトのすべてのプロパティを含めます。

デフォルト デフォルトの印刷オプションを選択するには、このボタンをクリックします。

5. 省略可能 : **プレビュー** をクリックして、結果情報がどのように印刷されるかを表示します。**プレビュー** ウィンドウが開きます。
6. 余白、向きなどのページ設定情報を変更するには、**プロパティ** をクリックします。
7. **OK** をクリックします。

ファイルへの結果の保存


テスト手順を文書化したり、他のテスト担当者と協業するために、ビジュアル テストまたはテスト スクリプトの結果を XML ファイル、タブ区切りファイル、またはプレーンなテキスト ファイルに保存することができます。ファイルに保存したら、それをレポート ツールにインポートして、表示、管理、フォーマット、印刷などを行えます。

 **注:** テキスト ファイルに保存したビジュアル テストまたはテスト スクリプトの結果をすばやく参照するには、ファイルをタブ区切りファイルとして Microsoft Excel にインポートします。結果が列見出しで整理されて表示されます。

1. ビジュアル テストまたはスクリプトの結果を開きます。
2. 以下の結果タブのいずれかをクリックします。
 - **要約**
 - **合格**
 - **失敗**
 - **フラグ**
 - **詳細**

選択したタブに表示されているデータのみが指定したファイルに保存されます。フィルタ処理されたデータは表示されず、結果の他のタブに存在しているデータは保存されません。

3. **ファイル > 名前を付けて保存** を選択します。 **名前を付けて保存** ダイアログ ボックスが開き、デフォルトの結果名が **ファイル名** テキスト ボックスに表示されます。

 **注:** デフォルトの結果名を変更するには、**再生結果** オプションの **デフォルトの結果名** オプションを変更します。

4. ファイルの保存場所を選択します。
5. ファイル名を入力するか、デフォルト名を使用します。
6. ファイルの種類を選択します。

任意のタブの結果を、XML 形式、タブ区切り形式、またはプレーンなテキスト形式で保存できます。

7. **保存** をクリックします。

Web ページ コンテンツのキャプチャ

ブラウザ ウィンドウに現在表示されている Web ページの一部をスクリーンショットにキャプチャする場合は、CaptureBitmap メソッドを使用できます。画像ファイルの場所と名前は、絶対パスまたは相対パスでパラメータで指定します。例：

```
'VB code
browserWindow.CaptureBitmap("C:¥Temp¥MyPage.png")
```

Web ページ コンテンツ全体のスクリーンショットを 1 つの画像としてキャプチャする場合は、CaptureFullPageBitmap メソッドを使用できます。画像ファイルの場所と名前は、絶対パスまたは相対パスでパラメータで指定します。例：

```
'VB code
browserWindow.CaptureFullPageBitmap("C:¥Temp¥MyPage.png")
```

結果フィルタ

多くの場合、結果には大量のデータが含まれます。最も重要なデータに焦点を合わせるために、合格/失敗のステータス、日付/時刻、フラグが付いたステップなどの基準に基づいて、あらかじめ定義したフィルタを適用するか、または専用のフィルタを定義できます。Silk Test Workbench は、現在表示されているビューに関係なく、**結果** ウィンドウの **合格**、**失敗**、**フラグ**、および **詳細** タブにフィルタを適用します。

結果フィルタの作成

結果フィルタを使用すると、**結果** ウィンドウに表示されるデータの量が減少するため、重要な結果をすばやく見つけることができます。フィルタを設定したら、**フィルタ** リストでそれを選択して適用します。

1. 結果を開きます。
2. ツールバーで **フィルタの管理** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。
3. **新規作成** をクリックします。**新しいフィルタ** ダイアログ ボックスが開き、デフォルトで **全般** タブが選択されます。
4. **名前** テキスト ボックスに、フィルタの名前を入力します。
ここで指定した名前がツールバーの **フィルタ** リストに表示されます。
5. 省略可能：**説明** テキスト ボックスにこのフィルタの使用方法的説明を入力します。
6. フィルタの種類を選択します。
 - このフィルタを現在のユーザーにのみ使用可能にするには **個人用** を選択します。
 - すべてのユーザーに使用可能にするには **公開用** を選択します。
7. ダイアログ ボックスの **フィルタ条件** セクションで、**結果** リストから以下のいずれかのオプションを選択します。
 - すべて** ビジュアル テストのすべてのステップ、またはスクリプトのすべてのコード行を表示します。
 - 失敗** 正しく実行されなかったステップまたはコード行のみを表示します。
 - 合格** 正しく実行されたステップまたはコード行のみを表示します。
8. **日付/時刻** オプションを選択します。
 - 任意の日付の任意の時刻に発生した結果データを表示するには、**すべて** を選択します。
 - 特定の時刻または日付に発生した結果データを表示するには、**開始** を選択して目的の日付/時刻の範囲を指定します。
9. **ユーザー** オプションを選択します。
 - すべてのユーザーによって作成された結果データを表示するには、**すべて** を選択します。

- 選択したユーザーによって作成された結果データのみを表示するには、**選択済み** を選択してから目的のユーザーを選択します。
- 10** 省略可能：フラグが割り当てられていないすべての結果データをフィルタ処理で非表示にするには、**フラグが割り当てられているステップのみ表示する** チェックボックスをオンにします。
- 11** より綿密なフィルタを作成するには、**詳細** タブをクリックし、必要な検証のみをフィルタ対象として指定します。
デフォルトでは、すべての検証が Silk Test Workbench によってフィルタの対象に含まれます。
- 左側のペインには、コマンドの種類のアとに、かっこで囲まれた数が表示されます。これらの数は、使用可能なコマンドの総数と選択されたコマンドの数を表します。たとえば、**検証 (3/7)** は、使用可能な 7 種類の検証から 3 種類の検証が選択されていることを示します。また、太字のチェックは、特定のコマンドの種類に関してすべてのコマンドが選択されていることを示します。オペレーティング システムの設定に応じて、チェックボックス内の四角、または無効になっているチェックが、特定のコマンドの種類に関して一部のコマンドが選択されていることを示します。
- 12** **OK** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。
- 13** アクティブな **結果** ウィンドウにフィルタを適用するには、フィルタを選択して **適用** をクリックし、**OK** をクリックします。
- 14** **閉じる** をクリックします。

結果フィルタの編集

フィルタを編集して、**結果** ウィンドウに表示されるデータを変更します。

1. 結果を開きます。
2. ツールバーで **フィルタの管理** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。
3. ユーザー定義フィルタを選択します。
デフォルトのフィルタ (**検証** と **失敗した検証**) は編集できません。
4. **編集** をクリックします。**フィルタの編集** ダイアログ ボックスが開き、デフォルトで **全般** タブが選択されます。
5. **名前** テキスト ボックスに、フィルタの名前を入力します。
ここで指定した名前がツールバーの **フィルタ** リストに表示されます。
6. 省略可能：**説明** テキスト ボックスにこのフィルタの使用法の説明を入力します。
7. フィルタの種類を選択します。
 - このフィルタを現在のユーザーにのみ使用可能にするには **個人用** を選択します。
 - すべてのユーザーに使用可能にするには **公開用** を選択します。
8. ダイアログ ボックスの **フィルタ条件** セクションで、**結果** リストから以下のいずれかのオプションを選択します。

すべて ビジュアル テストのすべてのステップ、またはスクリプトのすべてのコード行を表示します。

失敗 正しく実行されなかったステップまたはコード行のみを表示します。

合格 正しく実行されたステップまたはコード行のみを表示します。
9. **日付/時刻** オプションを選択します。
 - 任意の日付の任意の時刻に発生した結果データを表示するには、**すべて** を選択します。
 - 特定の時刻または日付に発生した結果データを表示するには、**開始** を選択して目的の日付/時刻の範囲を指定します。
10. **ユーザー** オプションを選択します。
 - すべてのユーザーによって作成された結果データを表示するには、**すべて** を選択します。
 - 選択したユーザーによって作成された結果データのみを表示するには、**選択済み** を選択してから目的のユーザーを選択します。

- 11 省略可能：フラグが割り当てられていないすべての結果データをフィルタ処理で非表示にするには、**フラグが割り当てられているステップのみ表示する** チェック ボックスをオンにします。
- 12 より綿密なフィルタを作成するには、**詳細** タブをクリックし、必要な検証のみをフィルタ対象として指定します。
デフォルトでは、すべての検証が Silk Test Workbench によってフィルタの対象に含まれます。
左側のペインには、コマンドの種類のとに、かっこで囲まれた数が表示されます。これらの数は、使用可能なコマンドの総数と選択されたコマンドの数を表します。たとえば、**検証 (3/7)** は、使用可能な 7 種類の検証から 3 種類の検証が選択されていることを示します。また、太字のチェックは、特定のコマンドの種類に関してすべてのコマンドが選択されていることを示します。オペレーティング システムの設定に応じて、チェック ボックス内の四角、または無効になっているチェックが、特定のコマンドの種類に関して一部のコマンドが選択されていることを示します。
- 13 **OK** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。
- 14 アクティブな **結果** ウィンドウにフィルタを適用するには、フィルタを選択して **適用** をクリックし、**OK** をクリックします。
- 15 **閉じる** をクリックします。

結果フィルタの適用

結果フィルタは、以下の 2 通りの方法で適用できます。結果フィルタは、アクティブな **結果** ウィンドウにのみ適用することも、開いているすべての **結果** ウィンドウに対して適用することもできます。

1. 結果を開きます。
2. 結果フィルタをアクティブな **結果** ウィンドウにのみ適用するには、**結果** ウィンドウのツールバーの **種類の選択によるフィルタ処理の結果** リストからフィルタを選択します。Silk Test Workbench によってフィルタがアクティブな **結果** ウィンドウに適用され、タイトル バーがフィルタの名前で更新されます。例：Silk Test Workbench - [Common - VisualTest1 (結果：フィルタ = 失敗した検証)] のように表示されます。
3. 結果フィルタを開いているすべての **結果** ウィンドウに適用するには、以下の手順を実行します。
 - a) 結果を開きます。
 - b) 適用するフィルタを選択します。
 - c) **適用** をクリックして、**OK** をクリックします。Silk Test Workbench によってフィルタがアクティブな **結果** ウィンドウに適用され、タイトル バーがフィルタの名前で更新されます。例：Silk Test Workbench - [Common - VisualTest1 (結果：フィルタ = 失敗した検証)] のように表示されます。結果フィルタを作成すると、**結果** ウィンドウのツールバーの **フィルタの管理** ツールバー ボタンのすぐ左にあるドロップダウン リストに表示されます。このリストからフィルタを選択すると、Silk Test Workbench は開いているすべての **結果** ウィンドウにそれを適用し、タイトルバーが更新されてフィルタの名前が表示されます。

結果フィルタの複製

フィルタの管理 ダイアログ ボックスを使用して、結果フィルタのコピーを作成できます。

1. 結果を開きます。
2. ツールバーで **フィルタの管理** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。
3. **フィルタ** リストから、複製するフィルタを選択します。
4. **複製** をクリックします。**フィルタの複製** ダイアログ ボックスが開きます。

適用された結果フィルタの削除

結果フィルタを削除して、**結果** ウィンドウにすべての結果データを表示します。

1. 結果を開きます。
2. ツールバーで **フィルタの管理** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。

3. **(なし)** を選択します。
4. **適用** をクリックします。
5. **閉じる** をクリックします。アクティブな **結果** ウィンドウからフィルタが削除されます。

結果フィルタの削除

フィルタを適用しないで結果を表示するには、ユーザー定義フィルタを削除します。

1. 結果を開きます。
2. ツールバーで **フィルタの管理** をクリックします。**フィルタの管理** ダイアログ ボックスが開きます。
3. 削除するユーザー定義フィルタを選択します。
ユーザーが定義したフィルタのみ削除できます。あらかじめ定義されているフィルタは削除できません。
4. **削除** をクリックし、**はい** をクリックして、フィルタの削除を確認します。
5. **閉じる** をクリックします。現在のユーザーの Silk Test Workbench データベースからフィルタが削除され、デフォルト フィルタ **なし** が開いているすべての結果に適用されます。

[条件] ダイアログ ボックス

条件 ダイアログ ボックスを使用して、今後すべてのテスト実行の合格条件として、ユーザー定義の検証の合格の割合を設定できます。

このダイアログ ボックスには、以下のオプションがあります。

合格の条件：今後すべての実行の合格条件として、検証の合格の割合を設定します。たとえば、合格条件を 90% に設定した場合、ビジュアル テストの 10 件の検証のうち 9 件に合格しないと、再生の結果が合格になりません。

また、このオプションを設定すると、再生結果の **結果合格条件** オプションも設定されます。


デフォルト：このボタンをクリックすると、合格条件がデフォルト値（100%）にリセットされます。

再生ステータス ダイアログ ボックスの有効化

再生ステータス ダイアログ ボックスを有効にすると、テストの再生中に実行される操作を表示できます。このダイアログ ボックスは、リモート Mac やモバイル デバイスなど、他のマシン上でテストを再生する場合に非常に役立ちます。

再生ステータス ダイアログ ボックスを有効化するには：

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。
3. **再生ステータス ダイアログ** をクリックします。
4. **再生ステータス** ダイアログ ボックスを有効化するには、**再生ステータス ダイアログの表示** オプションを **はい** に設定します。
5. **再生ステータス** ダイアログ ボックスにテスト対象アプリケーションのビデオまたはスクリーンショットを表示するには、**テスト対象アプリケーションの表示** オプションを **はい** に設定します。
6. **OK** をクリックします。

 **注**：リモート Mac やモバイル デバイス上でテストする場合は、Mac またはデバイスの画面がテスト中にオフにならないようにしてください。オフになると、**再生ステータス** ダイアログ ボックスには何も表示されなくなります。

[再生完了] ダイアログ ボックス

このダイアログ ボックスを使用して、再生が正常に完了したあとに実行する操作を指定できます。

次回からこのメッセージを表示しない チェック ボックスがオンになっているか、または **全般** オプションの **[再生完了] ダイアログの表示** ダイアログ オプションが **いいえ** に設定されている場合は、再生が正常に完了しても、このダイアログ ボックスは表示されません。この場合、再生が正常に完了すると、ビジュアル テストが **ビジュアル ナビゲータ** に表示され、スクリプトがスクリプト ウィンドウに表示されます。

再生完了 ダイアログ ボックスには、以下のオプションがあります。

もう一度再生する：ビジュアル テストまたはスクリプトを再生するには、このボタンをクリックします。このオプションを選択すると、再生が正常に完了したあと、**再生完了** ダイアログ ボックスが再び表示されます。

結果へ移動：**結果** ウィンドウを開くには、このボタンをクリックします。結果の表示と操作の詳細については、「結果の概要」を参照してください。

ビジュアル テストへ移動：ビジュアル テストの再生後にのみ表示されます。変更を保存せずに、**ビジュアル ナビゲータ** で現在のビジュアル テストを開くには、このボタンをクリックします。

.NET スクリプトへ移動：スクリプトの再生後にのみ表示されます。変更を保存せずに、スクリプト ウィンドウで現在のテスト スクリプトを開くには、このボタンをクリックします。

保存：変更を保存したあとで現在のビジュアル テストまたはテスト スクリプトを表示するには、このボタンをクリックします。

次回からこのメッセージを表示しない：このチェック ボックスをオンにすると、**再生** ダイアログ ボックスが今後の再生で表示されなくなります。**全般オプション** の **[再生完了] ダイアログの表示** を **はい** に設定すると、このオプションを再度選択できるようになります。



ヒント：**記録完了** ダイアログ ボックスで **次回からこのメッセージを表示しない** チェック ボックスがオンになっている場合、このダイアログ ボックスは表示されません。この場合、ビジュアル テストが記録されたときは **ビジュアル ナビゲータ** が表示され、スクリプトの場合はスクリプト ウィンドウが表示されます。

[再生エラー] ダイアログ ボックス

このダイアログ ボックスを使用して、再生エラーが発生したときに実行する操作を指定できます。

このダイアログ ボックスは、ビジュアル テストまたはスクリプトの再生中に Silk Test Workbench でエラーが発生した場合に表示されます。

再生エラー ダイアログ ボックスには、以下のボタンがあります。

続行：エラーが発生したステップを再試行して、再生を再開します。エラーは、ビジュアル テストの再生結果の **詳細** タブに表示されます。エラーの解決後、このオプションを使用して、再生を再開します。このボタンは、ビジュアル テストでのみ有効になります。

終了：エラーが発生したステップで再生を停止します。**再生完了** ダイアログ ボックスが表示され、ビジュアル テストまたはスクリプトを再生したり、再生結果を表示したり、ビジュアル テストまたはスクリプトに戻ったりすることができます。

デバッグ：デバッグ モードに切り替えます。デバッグ モードでは、再生を段階的に実行しながら、ビジュアル テストのエラーの診断や修正を行うことができます。このボタンは、ビジュアル テストでのみ有効になります。

ヘルプ：オンライン ヘルプ内のこのヘルプ トピックを表示します。

資産の管理

資産はテストプロジェクトの基本要素です。資産とは、定義してデータベースに保存するテスト要素のことです。資産は **アセットブラウザ** を使用して表示および管理できます。

ツールバー ボタンおよび **アセットブラウザ** を使用して資産を管理します。**開始画面** の **入門ガイド** ペインに移動して、ペインの **タスク** 領域の **アセットブラウザ** をクリックします。**アセットブラウザ** では、テストプロジェクトの資産を一元的に作成、管理、表示できます。

ほとんどの資産は、それを参照している他の資産を変更せずに、再利用や再定義ができます。

資産の種類

Silk Test Workbench は、次の種類の資産をサポートします。

ビジュアルテスト ビジュアルテストは、アプリケーションのテスト時にスクリーンショットを使用してユーザーの操作を模倣する、自動化されたテストです。ビジュアルテストは、自動テストソリューションの基本要素です。

ビジュアルテストの作成方法についての情報は、「[ビジュアルテストの作成](#)」を参照してください。

.NET スクリプト VB .NET スクリプトも、アプリケーションのテスト時にユーザーの操作を模倣する、自動化されたテストです。このスクリプトは、Microsoft Visual Basic .NET で作成されています。

VB .NET スクリプトの作成方法についての情報は、「[スクリプトの作成](#)」を参照してください。

キーワードテスト キーワード駆動テストは、実行可能なキーワードのコレクションです。キーワード駆動テストは、他のテストと同様に再生することができます。

キーワード駆動テストについての追加の情報は、「[キーワード駆動テスト](#)」を参照してください。

キーワード キーワードは、テストオブジェクトに対する複数の操作の組み合わせを定義したものです。キーワードの実装は、さまざまなツールとプログラム言語 (Java や .NET など) を使用して行えます。Silk Test Workbench でのキーワードは、ビジュアルテストまたはメソッド名の前に Keyword 属性を持つメソッドです。キーワードは、キーワード資産として保存されます。

キーワードシーケンスは、他のキーワードを組み合わせたものです。キーワードシーケンスは、頻繁に使用するキーワードの組み合わせを 1 つのキーワードにまとめることにより、メンテナンスの労力を低減し、テストを理解しやすくすることができます。

キーワードについての追加の情報は、「[キーワード](#)」を参照してください。

結果 結果は、ビジュアルテストまたは .NET スクリプトの再生時に生成される資産です。結果は、テストプロセスを証明し、任意の時点におけるテストアプリケーションの機能の状態を永続的に記録します。

結果についての追加の情報は、「[結果の分析](#)」を参照してください。

オブジェクトマップ オブジェクトマップには、論理名とコントロールおよびウィンドウを関連付ける項目が格納されています。ロケータの代わりにこれらの論理名を使用して、アプリケーションのコントロールおよびウィンドウを識別できます。



注: オブジェクトマップ項目を含むオブジェクトマップの名前を変更すると、ルート項目名が変更されます。このオブジェクトマップを使用する既存のビジュアルテストも

またはスクリプトを、新しいオブジェクト マップのルート項目名を使用するように手動で変更する必要があります。

オブジェクト マップについての追加の情報は、「[オブジェクト マップ](#)」を参照してください。

イメージ検証 イメージ検証は、イメージがテスト対象アプリケーションの UI に含まれているかどうかをチェックします。

イメージ検証についての追加の情報は、「[イメージ検証](#)」を参照してください。

アクティブデータ アクティブ データは、ビジュアル テストでデータ ドリブン テストを作成するために使用されます。アクティブ データによって、強力な包括的なアプリケーション テスト ソリューションの入力として外部ファイルの既存のデータを利用できます。トランザクションごとに異なるデータセットを使用することにより、複雑なコードを記述したり、既存のデータを損なうことなく、テスト アプリケーションで複数のトランザクションを実行できます。

アクティブ データについての追加の情報は、「[アクティブ データ](#)」を参照してください。

イメージ資産 イメージ資産は、データベースに資産として格納されるイメージです。イメージ資産を使用して、Silk Test Workbench で認識されないコントロールをクリックできるようになります。

イメージ資産についての追加の情報は、「[イメージ資産](#)」を参照してください。

資産のバージョンをパージ

Silk Test Workbench は、資産を保存するたびに新しいバージョンを作成します。したがって、資産のバージョンの数が増加し、データベースのサイズも大きくなります。

データベースから資産のバージョンを削除するには、**資産のバージョンをパージ** ダイアログ ボックスを使用して、資産の種類、資産が属するプロジェクト、保持するバージョンの数、資産の変更日などの条件を指定します。



注: 資産の現在のバージョンは削除できません。この機能は、資産を完全に削除するためのものではなく、資産のバージョンのみを削除するものです。資産のバージョンをパージするには、スクリプト作成者かフル アクセス権限が必要です。

資産のバージョンをパージ

資産のバージョンをパージする前に、ログイン中のユーザーが必要なアクセス権を持っていることを確認してください。パージの実行に必要なアクセス権は、グローバル オプションで変更できます（管理者権限が必要）。実行に必要なアクセス権の変更についての情報は、[グローバル オプション](#) を参照してください。

1. **ファイル > 資産のバージョンをパージ** をクリックします。 **資産のバージョンをパージ** ウィザードが表示されます。
2. **次へ** をクリックします。
3. 資産をパージする方法を選択します。
 - 特定の種類の資産をパージするには、**資産の種類によるパージ** をクリックします。
 - 個々の資産をパージするには、**個々の資産のパージ** をクリックします。
4. [資産の種類によるパージ] を選択した場合は、次の手順に従います。
 - a) 資産のバージョンをパージする資産の種類をチェックします。
 - b) 選択した資産の種類のバージョンをパージするプロジェクトをチェックします。
5. [個々の資産のパージ] を選択した場合は、次の手順に従います。
 - a) プロジェクト ツリーで、バージョンをパージする資産を含むプロジェクト ノードを展開します。
 - b) 資産の種類をクリックします。
 - c) リストから個々の資産をチェックします。

6. 省略可能 : [資産のバージョンをパージ] のプロジェクトの選択状態を保持 オプションをオンにすると、資産のバージョンをパージ ウィザードで選択したプロジェクトを保持できます。

a) メニューから ツール > オプション を選択します。

b) オプション メニューで、グローバル > 資産管理 を選択します。

c) [資産のバージョンをパージ] のプロジェクトの選択状態を保持 オプションを はい に設定します。

このオプションを いいえ に設定すると、資産のバージョンをパージ ウィザードを開いたときに、プロジェクトは選択されません。

7. 次へ をクリックします。

8. 指定した数の資産のバージョンを保持するか、指定した日より前に変更されたすべての資産のバージョンを削除するかを選択します。

9. 指定した数の資産のバージョンを保持したい場合は、保持する資産のバージョンの数を指定します。

指定した数よりも古い資産のバージョンがデータベースから削除されます。たとえば、2 を指定すると、最新の 2 つのバージョンが維持され、残りのバージョンが削除されます。維持するバージョンの最低数は 1 で、最大数は 32767 です。



注: すべての実際の検証はパージ中に維持され、結果が削除されるときに削除されます。結果によって使用されている予想した検証も削除されません。パージ後に、指定したものより多くの予想した検証が保持される可能性があります。たとえば、予想した検証の 5 バージョンがテスト結果によって参照されており、パージ後に各資産ごとに 3 バージョンのみ残したい場合、結果によって参照されている検証の 5 バージョンすべてが保持されます。

10 指定した日より前に変更されたすべての資産のバージョンを削除したい場合は、定義済みの日付を選択するか、1970/01/01 から 3000/12/31 の間の日付を指定します。



注: すべての実際の検証はパージ中に維持され、結果が削除されるときに削除されます。結果によって使用されている予想した検証も削除されません。これによって、指定した日付よりも古い予想した検証が残されるという状況が生じる可能性があります。

11 次へ をクリックします。資産のバージョンをパージ ウィザードは、削除する資産のバージョンの要約を表示します。

12 次へ をクリックします。



注: 資産または複数の資産をパージできなかった場合は、メッセージ ボックスにパージできなかった資産の数が示されます。通常、これは別のユーザーが資産の現在のバージョンを開いているか、パージ中にその資産がすでに削除されていることが原因です。


13 完了 をクリックします。


パージのコマンド ライン パラメータ

GUI を使用する代わりに、コマンド ラインを使用して資産のバージョンをパージできます。

コマンド ライン プログラムの名前は STWPurge.exe であり、次の場所にあります : %Silk%SilkTest%ng %gui%。

パラメータ	構文	例	説明
ユーザー名	u	-u admin	データベースへの接続を許可されているユーザーの名前を指定します。このパラメータは必須です。
	username	/username JohnSmith	
パスワード	p	-p admin123	データベースへの接続を許可されているユーザーの大文字小文字が区別されるパスワードを指定します。このパラメータはパスワードが Nothing 以外の場合に必要です。
	password	/password admin123	

パラメータ	構文	例	説明
DSN	d dsn	-d mydsn /dsn mydsn	データベースへの接続に使用するデータソース名 (DSN) の名前を指定します。このパラメータは省略可能です。省略した場合は、デフォルトの DSN が使用されます。
資産の種類	s assettypes	-s visualtest script /assettypes visualtest	<p>資産バージョンを削除する資産の種類を指定します。このパラメータは省略可能です。省略した場合は、すべての資産の種類が対象となります。このパラメータは、以下の値を受け入れます (構文はカギ括弧で示します、複数値は空白で区切って指定できます)。</p> <ul style="list-style-type: none"> • ビジュアル テスト [visualtest] • オブジェクト マップ [objectmap] • 結果 [result] • アクティブデータ [activedata] • .NET スクリプト [script]
プロジェクト	r prj projects	-r commonprojectA /projects common projects	<p>資産のバージョンを削除するプロジェクト、資産の種類、個々の資産を指定します。このパラメータは省略可能です。省略した場合は、すべてのプロジェクトが対象となります。複数の項目を指定する場合は、空白で区切って入力します。キャレット記号 (^) を含む項目は、引用符で囲む必要があります。</p> <p>次の例は、プロジェクトパラメータを使用して、<i>Login</i> という名前のビジュアルテストから古い資産のバージョンをパージします。</p> <pre>-projects Common.VisualTest.Login</pre> <p>次の例は、プロジェクトパラメータを使用して、<i>MyProject</i> という名前のプロジェクトのすべてのアクティブデータ資産から古い資産のバージョンをパージします。</p> <pre>-projects MyProject.ActiveData</pre> <p>projects パラメータを使用して資産の種類や個々の資産を指定する場合は、assettypes パラメータは指定できません。ただし、プロジェクトを指定する場合は同時に指定することができます。例：</p> <pre>-projects Common -assettypes script</pre>
数によるパージ	pc purgebycount	-pc 10 /purgebycount	<p>指定された数を超える古い資産バージョンは削除されます。このパラメータまたは日付によるパージパラメータのいずれかが必要です。両方のパラメータを使用することはできません。維持するバージョンの最低数は 1 で、最大数は 32767 です。資産のアクティブなバージョンは削除できません。このパラメータは、単独の数値を受け入れます。たとえば、10 を指定した場合、資産の 10 の最新バージョンが維持され、それより古いバージョンは削除されます。</p> <p> 注: すべての実際の検証はパージ中に維持され、結果が削除されるときに削除されます。結果によって使用されている予想した検証も削除され</p>

パラメータ	構文	例	説明
			<p>ません。ページ後に、指定したものより多くの予想した検証が保持される可能性があります。たとえば、予想した検証の5バージョンがテスト結果によって参照されており、ページ後に各資産ごとに3バージョンのみ残したい場合、結果によって参照されている検証の5バージョンすべてが保持されます。</p>
日付による ページ	pd purgebydate	-pd 2011-03-28 /purgebydate 2012-03-28	<p>指定された日付以前に変更された資産バージョンを削除します。このパラメータまたは数によるページパラメータのいずれかが必要です。両方のパラメータを使用することはできません。このパラメータは、1970-01-01 から 3000-12-31 までの日付を次の形式で受け入れます：[YYYY-MM-DD]。</p> <p> 注: すべての実際の検証はページ中に維持され、結果が削除されるときに削除されます。結果によって使用されている予想した検証も削除されません。これによって、指定した日付よりも古い予想した検証が残されるという状況が生じる可能性があります。</p>
情報モード	i info	-i silent /info verbose	<p>プログラムの実行中に表示する情報の種類を設定します。このパラメータは省略可能です。省略した場合は、デフォルトモードの「通常」が使用されます。このパラメータは、以下の値を受け入れます (構文はカギ括弧で示します)。</p> <p>通常 [n normal] すべてのエラーと一般情報と出力します。</p> <p>サイレント [s silent] 実行が停止するエラーのみを出力します。</p> <p>詳細 [v verbose] すべての警告、エラー、および一般情報と出力します。</p>
確認	o prompt	-o /prompt	<p>資産ページレポートを表示した後に、確認メッセージを表示します。このパラメータは省略可能です。</p>
ヘルプ	h ? help	-h /help	<p>ヘルプの内容を表示し、その他のすべてのパラメータを無視します。</p>

例 1

この例は、*Common* プロジェクトのすべての結果から、結果ごとに5バージョンを残してページします。

```
stwpurge /dsn SilkTest /username admin /password admin /purgebycount 5 /projects Common /assettypes result
```

例 2

この例は、*LoginTest* ビジュアルテストから、2017年12月31日より前に作成されたすべてのバージョンをパージします。

```
stwpurge /dsn SilkTest /username admin /password admin /purgebydate  
2016-12-31 /projects Common.VisualTest.LoginTest
```

アセットブラウザを使用した資産の作成

ほとんどの資産は、既存のビジュアルテストまたはスクリプトに挿入することによって作成されるか、または記録時に作成されます。ただし、**アセットブラウザ** を使用して資産を作成することもできます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面の入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストで適切な資産の種類を選択し、資産のリストを表示します。
3. 資産のリストで、作成する資産を右クリックします。この種類の資産を新規作成するには、**新規作成 > 資産の種類** を選択します。

すべての資産の種類をリストするには、**新規作成** を選択します。

すべての資産は、資産の命名規則に従う必要があります。

新しい資産は、現在アクティブになっているプロジェクトに追加され、そのプロジェクト内の他の資産に対して、またはアクティブプロジェクトを参照しているプロジェクトの資産に対してのみ使用可能になります。アクティブプロジェクトを変更する場合は、ウィンドウの**プロジェクト**領域の**アクティブ**リストから目的のプロジェクトを選択します。複数のプロジェクトで使用する資産を格納するには、Commonプロジェクトを使用します。Commonプロジェクトの資産は、Commonプロジェクトを参照しているすべてのプロジェクトで使用できます。

スクリプトから資産を開く

スクリプトを編集しているときに、資産を右クリックして**資産を開く**を選択し、資産を開くことができます。これにより、GUIで資産が開きます。

資産がシステム上のファイルへの参照である場合 (ImageClickFile によって参照される場合など)、ファイルはシステムのデフォルトエディターで開かれます。

資産を開く

アセットブラウザ を使用して、資産を開きます。

1. **ファイル > 開く** を選択するか、**開始画面の入門ガイド** ペインに移動して、**タスク**領域の**アセットブラウザ** をクリックします。
2. **資産の種類** リストで適切な資産の種類を選択し、資産のリストを表示します。
3. 右側のペインで、開く資産をダブルクリックします。

資産の比較

同じ種類の2つの異なる資産や、同じ資産の2つのバージョンを比較して、資産間や資産のバージョン間の相違点を調べることができます。


資産または資産のバージョン間の相違点のマージについては、「[資産のマージ](#)」を参照してください。

1. **表示** > **アセットブラウザ** を選択するか、**開始画面** の **入門ガイド** > **アセットブラウザ** を選択します。
2. **資産の種類** リストから資産の種類を選択します。

次の資産の種類に対して、資産や資産のバージョンを比較できます。

- ビジュアル テスト
- .NET スクリプト
- オブジェクト マップ


選択したプロジェクトの選択した種類の既存の資産が、右側のペインに表示されます。

 **ヒント:** 調査したい資産が表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 同じ種類の 2 つの異なる資産を比較するには：
 - a) **Ctrl + クリック** を使用して、比較する 2 つの資産を選択します。
 - b) 選択した項目を右クリックします。
 - c) **相違点の表示** をクリックします。
4. 同じ資産の 2 つのバージョンを比較するには：
 - a) 調査したい資産を右クリックして、**すべてのバージョンを表示** を選択します。
 - b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する 2 つのバージョンを選択します。
 - c) **相違点の表示** をクリックします。

Silk Test Workbench は、次の相違点を表示します。

- 変更箇所は黄色で表示されます。
- 追加箇所は緑色で表示されます。
- 比較した資産の一方に存在しない項目はグレーで表示されます。

 **注:** Silk Test Workbench ユーザーが資産のアクティブバージョンを比較またはマージしている間、その資産はロックされ、他のユーザーは編集できなくなります。資産が他のウィンドウで編集用に既に開かれている場合に、その資産のアクティブバージョンを比較またはマージしようとする、読み取り専用モードでそのバージョンが開かれます。読み取り専用かどうかは、そのバージョンのタイトルバーに表示されます。

資産のマージ


同じ種類の 2 つの異なる資産や、同じ資産の 2 つのバージョンを比較して、資産間や資産のバージョン間の相違点をマージすることができます。

1. **表示** > **アセットブラウザ** を選択するか、**開始画面** の **入門ガイド** > **アセットブラウザ** を選択します。
2. **資産の種類** リストから資産の種類を選択します。

次の資産の種類に対して、資産や資産のバージョンの相違点をマージできます。

- ビジュアル テスト
- .NET スクリプト
- オブジェクト マップ

選択したプロジェクトの選択した種類の既存の資産が、右側のペインに表示されます。

 **ヒント:** 調査したい資産が表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 同じ種類の 2 つの異なる資産を比較するには：
 - a) **Ctrl + クリック** を使用して、比較する 2 つの資産を選択します。
 - b) 選択した項目を右クリックします。
 - c) **相違点の表示** をクリックします。

4. 同じ資産の 2 つのバージョンを比較するには：
 - a) 調査したい資産を右クリックして、**すべてのバージョンを表示** を選択します。
 - b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する 2 つのバージョンを選択します。
 - c) **相違点の表示** をクリックします。
5. 相違点の表示を移動するには、ツールバーの **次の相違点** と **前の相違点** を使用します。
6. 選択した相違点をマージするには：
 - **左側にコピー** をクリックすると、右側のペインに表示されているバージョンまたは資産の選択した相違点を、左側に表示されているバージョンまたは資産にマージします。
 - **右側にコピー** をクリックすると、左側のペインに表示されているバージョンまたは資産の選択した相違点を、右側に表示されているバージョンまたは資産にマージします。

Silk Test Workbench は、次の相違点を表示します。

- 変更箇所は黄色で表示されます。
- 追加箇所は緑色で表示されます。
- 比較した資産の一方に存在しない項目はグレーで表示されます。



注: Silk Test Workbench ユーザーが資産のアクティブバージョンを比較またはマージしている間、その資産はロックされ、他のユーザーは編集できなくなります。資産が他のウィンドウで編集用に既に関われている場合に、その資産のアクティブバージョンを比較またはマージしようとすると、読み取り専用モードでそのバージョンが開かれます。読み取り専用かどうかは、そのバージョンのタイトルバーに表示されます。

資産の複製

複製した資産には、元の資産のすべての属性が保持されます。資産を複製することにより、似たような属性の同じ種類の資産が複数必要な場合に、時間を節約することができます。資産を作成して保存したあとは、いつでもそれを複製できます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面 の 入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストで適切な資産の種類を選択し、資産のリストを表示します。
3. **アセットブラウザ** の右側のペインで、複製する資産を右クリックし、**複製** を選択します。**資産の複製** ダイアログ ボックスが開きます。
4. 複製資産の名前を入力し、必要に応じて資産を保存するプロジェクトを変更し、新しい資産の説明を入力します。
5. **OK** をクリックします。

資産名の変更

作成して保存した資産は、いつでも名前を変更できます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面 の 入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストで適切な資産の種類を選択し、資産のリストを表示します。
3. **アセットブラウザ** の右側のペインで、名前を変更する資産を右クリックし、**名前の変更** を選択します。名前を変更したい資産を選択して、**F2** を押しても変更できます。**資産名の変更** ダイアログ ボックスが開きます。
4. 資産の新しい名前と、説明を入力します。
5. 省略可能：資産を他のプロジェクトに移動する場合は、資産の移動先のプロジェクトを選択します。ビジュアル テストや VB .NET スクリプトを他のプロジェクトに移動する場合に、この機能を使用できます。

6. 省略可能：資産の説明を入力します。
7. OK をクリックします。
8. 資産が他の資産から参照されている場合は、その処理方法を選択します。
 - 資産名を変更して、参照を更新する場合は、**この資産の名前を変更して参照を更新する** をクリックします。
 - 参照を更新せずに資産名を変更する場合は、**この資産の名前のみ変更する** をクリックします。



注: 参照している資産は、正しく再生できなくなる可能性があります。

- 資産名の変更をキャンセルする場合は、**キャンセル** をクリックします。

資産を別のプロジェクトに移動する場合に、新しいプロジェクトが参照している資産のプロジェクトから参照されていないと、参照している資産にエラー アイコンが表示されます。エラー アイコンにマウスカーソルを移動させると、エラーの情報が表示されます。

資産の削除

資産を作成して保存したあとは、いつでもそれを削除できます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面 の 入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** で適切な資産の種類を選択します。
3. 資産のリストで削除する資産を右クリックし、**削除** を選択します。**削除** ダイアログ ボックスが開きます。資産に複数のバージョンがある場合は、Silk Test Workbench により最新のバージョンを削除するかすべてのバージョンを削除するかのオプションが表示されます。
4. 資産に複数のバージョンがあり、そのすべてを削除する場合は、**すべてのバージョンを削除** を選択して**はい** をクリックします。複数のバージョンがない場合、現在（つまり唯一）のバージョンの資産を削除するには**はい** をクリックし、資産を削除せずに**削除** ダイアログ ボックスを閉じるには**いいえ** をクリックします。

資産リストのフィルタ処理

フィルタ リストを使用すると、特定の条件を満たす資産だけを表示できます。

1. **アセットブラウザ** ウィンドウで、**フィルタ** をクリックします。**フィルタ** ダイアログ ボックスが開きます。
2. 使用するフィルタを 1 つまたは複数指定します。
複数のフィルタを指定した場合、すべての条件を満たす資産が表示されます。以下は、フィルタの種類です。

作成者 ユーザーの名前を選択すると、そのユーザーによって作成されたすべての資産が表示されます。

作成日 開始日と終了日を指定すると、その期間中に作成されたすべての資産が表示されます。

最終更新者 ユーザーの名前を選択すると、そのユーザーによって更新されたすべての資産が表示されます。

最終更新日 開始日と終了日を指定すると、その期間中に更新されたすべての資産が表示されます。

3. **アセットブラウザ** で特定の名前を持つ資産のみを表示するには、**名前フィルタ** テキスト ボックスに名前を入力します。

アセットブラウザ には、**名前フィルタ** と他の任意のフィルタ条件に一致する資産が表示されます。

4. **OK** をクリックします。Silk Test Workbench により、現在表示されている資産項目のリストにフィルタが適用されます。



注: フィルタを解除して、再びすべての項目を表示するには、**フィルタ** をクリックし、**フィルタ** ダイアログ ボックスで **デフォルト** をクリックします。

アセットブラウザ で特定の名前を持つ資産のみを表示するには、**名前フィルタ** テキスト ボックスに名前を入力します。他の任意のフィルタも表示される資産セットに影響を与えます。

資産の検索

アセットブラウザ の **名前フィルタ** フィールドを使用して、名前を指定して資産を検索できます。

1. **アセットブラウザ** を開きます。
2. 資産の種類を選択します。
3. **名前フィルタ** フィールドをクリックします。
代わりに、**F3** を押すこともできます。
4. 検索する資産の名前を指定します。
資産の名前を入力していくと、**名前フィルタ** フィールドの文字に一致する資産だけが Silk Test Workbench に表示されます。

資産の並べ替え

アセットブラウザ の項目は、列の内容に従って並べ替えることができます。デフォルトでは、資産のリストは昇順に並んでいます。

1. **アセットブラウザ** に表示された項目の並べ替えに使用する列の見出しをクリックします。
たとえば、項目を日付順に並べるには、**作成日** 列の見出しをクリックします。
2. 並べ替えの方向を逆にするには、**アセットブラウザ** に表示された項目の並べ替えに以前使用した列の見出しをクリックします。
たとえば、作成日による並べ替えの順序を逆にするには、**作成日** 列の見出しを 2 回クリックします。

資産保存時のデフォルト動作の設定

資産を保存するときの、デフォルトの保存動作を設定します。動作は、資産の新しいバージョンを作成する、または資産の現在のバージョンを置き換える、のいずれかから選択します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **グローバル** の隣にあるプラス記号 (+) をクリックし、**資産管理** をクリックします。**資産管理** オプションが右側のパネルに表示されます。
3. **デフォルトの保存動作** リスト ボックスから、**新しいバージョンとして保存** または **現在のバージョンとして保存** を選択します。ツールバーのボタンやショートカット キー (Ctrl+S) で資産を保存するとき、Silk Test Workbench によって、資産の新しいバージョンが作成されるか、資産の現在のバージョンが置き換えられます。
4. **OK** をクリックします。


資産のバージョンの最大数の設定

テスト結果を含む資産のストレージの容量を制限するために、保持する資産のバージョンの最大数を指定できます。

1. メニューで、**ツール > オプション** を選択します。**オプション** ダイアログ ボックスが開きます。

2. **オプション** ツリーで、**グローバル** を展開します。
3. **資産管理** を選択します。
4. **資産管理** ページの **資産のバージョンの最大数** テキスト ボックスに保持するバージョンの数を入力します。
デフォルトの設定は 0 で、すべての資産のバージョンが保存されることを意味します。
5. **OK** をクリックします。

バージョンの最大数が設定され、その数に到達すると、資産の新しいバージョンが保存され、最も古い資産のバージョンが削除されます。資産をインポートする場合、バージョンの数が最大数を超過していたとしても、すべてのバージョンがインポートされます。このようにしてインポートされた資産に対して新しいバージョンが保存されるとき、指定した資産の最大数と一致するように、最も古いバージョンの資産から削除されます。

 **注:** 資産のバージョンの数を制限すると、古いテスト結果は自動的に削除されます。

インクルードする資産の表示

他の複数のテスト資産のテスト資産を再利用することができます。たとえば、ビジュアル テストや VB .NET スクリプトで同じオブジェクト マップを使用できます。各スクリプトには、使用する他のテスト資産を明示的に含める必要があります。Silk Test Workbench は VB .NET 以外の資産に対しては自動的にインクルード参照を作成しますが、VB .NET 資産に対しては、それを使用する各スクリプトにユーザーが明示的に含める必要があります。

インクルード情報 ペインを使用して、選択した資産に含まれる資産と選択した資産を含む資産を表示できます。たとえば、スクリプトを編集したときに、このペインを使ってスクリプトを利用するほかのすべてのスクリプトを探して実行し、動作することを確認できます。

また、オブジェクト マップを変更したときにも、**インクルード情報** ペインを使用できます。たとえば、オブジェクト マップ ID の名前を変更する場合、Silk Test Workbench は、そのオブジェクト マップを使用するスクリプトで ID を自動的に更新しません。**インクルード情報** ペインを使うと、変更したオブジェクト マップ ID に必要なスクリプトを探し出すことができます。

1. **アセットブラウザ** で資産 を選択します。
2. **表示 > インクルード** を選択します。 **インクルード情報** ペインが開きます。
3. **インクルード情報** ペインで、**インクルード** タブをクリックして選択した資産に含まれる資産を表示するか、**インクルード元** タブをクリックして選択した資産を含む資産を表示します。

複数バージョンの資産の表示

Silk Test Workbench では、すべての種類の資産のバージョンが自動的に格納されます。資産を変更するたびに、その資産の新しいバージョンがデータベースに格納されます。資産のすべてのバージョンのリストと、資産を最後に更新したユーザー、更新日時、バージョン番号などのバージョン情報をいつでも表示できます。

Silk Test Workbench では、資産のアクティブ バージョンだけを参照します。

1. **表示 > アセットブラウザ** を選択するか、**開始画面** の **入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストで適切な資産の種類を選択し、資産のリストを表示します。
3. 資産のリストで表示する資産を右クリックし、**すべてのバージョンを表示** を選択します。**バージョンの詳細** ダイアログ ボックスが開きます。
4. **バージョンの詳細** ダイアログ ボックス下部に並んでいるボタンを使用して、資産のバージョンを開いたり、複製したり、削除したり、閉じたりできます。

- バージョンを選択し、**アクティブバージョンにする** をクリックすると、任意のバージョンを現在アクティブなバージョンにできます。

資産の命名規則

資産の新規作成 ダイアログ ボックスで資産を追加すると、Silk Test Workbench によって名前がチェックされ、所定の命名規則に従っているかどうかを検証されます。

Silk Test Workbench では、以下の命名規則が使用されます。

- 資産名の長さは 1 から 126 文字である必要があります。
- 資産名に ¥ / < > " ' : * ? | = の各文字を使用することはできません。
- 同一プロジェクト内で同じ種類（またはグループ）の他の資産と同じ名前を付けることはできません。

テスト資産のインポートとエクスポート

インポートおよびエクスポートユーティリティを使用して、異なるデータベース間で資産を転送できます。資産のインポートおよびエクスポートによって、プロジェクト管理が簡単になります。また、WAN を介して資産を共有したり、外部のバージョン管理システムを活用したりできます。資産をインポートまたはエクスポートするには、**資産インポート** ウィザードまたは **資産エクスポート** ウィザードを使用できます。いずれの方法でも、必要に応じて資産のインポートとエクスポートをカスタマイズできます。

インポートおよびエクスポートが可能な資産は、プロジェクト資産とシステム資産です。

プロジェクト資産 詳細については、「資産の種類」を参照してください。

システム資産 記録と再生のオプション。

資産のインポート

資産のデータベース間でのインポートは、**資産インポート** ウィザードを使用して行います。

1. **ファイル > 資産のインポート** を選択します。 **Silk Test Workbench 資産インポート** ウィザードが開きます。

このウィザードおよびそれ以外のすべての Silk Test Workbench ウィザードを今後使用するとき、**よろこ** ページをスキップするには、このページの最下部にあるチェックボックスをオンにします。このチェックボックスをオンにすると、**全般** オプションの **ウィザードの [よろこ] 画面の表示** オプションが更新されます。

ウィザードのサイズは、任意のページの右下隅をドラッグして変更できます。

2. **次へ** をクリックします。 **インポート ファイル名** ページが開きます。
3. インポートする資産を含んだ ZIP ファイルを指定します。
4. **次へ** をクリックします。 **インポートする項目** ページが開きます。
5. 以下のいずれかのステップを実行して、インポートする項目を選択します。

- すべての資産をインポートするには、**すべての資産をインポートする** をクリックします。

資産のすべてのバージョンをインポートするには、**各資産の最新バージョンだけをインポートする** チェックボックスをオフにします。


- 特定の資産をインポートするには、**選択した資産だけをインポートする** をクリックします。

各資産の特定のバージョンをインポートするには、**各資産のすべてのバージョンを表示する** チェックボックスをオンにします。その後、**次へ** をクリックします。

インポートする資産のチェックボックスをオンにします。


6. **次へ** をクリックします。 **プロジェクトの選択** ページが開きます。

7. **省略可能** : ソースプロジェクトの資産をインポートするターゲットプロジェクトを指定するには、**ターゲット** 列でプロジェクトを選択します。

 **注:** 現在のユーザーが管理者であれば、新しいプロジェクトを作成する項目を **ターゲット** リストから選択して、資産を新しいプロジェクトにインポートできます。

ターゲット リストにプロジェクトが含まれていない場合は、次の原因が考えられます。

- インポートファイルのプロジェクトがデータベースに存在せず、現在のユーザーが管理者でない
- 現在のユーザーが、既存のプロジェクトに対して実行者以上の権限を持たない

 **注:** 管理者であれば、新しいプロジェクトを作成して資産をインポートすることができます。

8. **次へ** をクリックします。 **競合の解決** ページが開きます。

9. インポートする資産名が既存の資産名と競合しているプロジェクト資産について、競合の解決方法を選択します。

次のいずれかを選択します。

- **インポートした資産を既存の資産の最新バージョンとして追加する** : このオプションは、データベース内の資産の新しいバージョンを作成します。インポートされる資産のバージョンが複数ある場合、資産は最小のバージョンが先に、最大のバージョンが最後にインポートされます。つまり、最大のバージョン番号を持つ資産がデータベースの最新の資産バージョンになります。
- **インポートした資産を追加、名前変更、またはスキップするか確認する** : これは対話オプションで、各競合に対してダイアログボックスを生成して、インポート資産を追加するか、名前変更するか、またはスキップするよう要求します。
- **資産をインポートしない** : 資産のインポートをスキップします。

10. インポートする資産名が既存の資産名と競合しているシステム資産について、競合の解決方法を選択します。

次のいずれかを選択します。


- **インポートした資産を名前変更またはスキップするか確認する** : これは対話オプションで、各競合に対してダイアログボックスを生成して、インポート資産を名前変更するか、またはスキップするよう要求します。
- **資産をインポートしない** : 資産のインポートをスキップします。

11. **次へ** をクリックします。 **資産のインポートの開始** ページが開きます。

12. 要約を確認します。

資産のインポートでは、次の点にご注意ください。


- ZIP ファイルに含まれるプロジェクトのうち、ターゲットデータベースに存在しないプロジェクトが作成されます。
- データベースに既に存在するプロジェクトには、プロジェクト参照は追加されません。要約には、このような競合が表示されます。必要に応じて、プロジェクト間の参照を手動で追加してください。

 **注:** 19.0 より前のバージョンの Silk Test Workbench で作成した ZIP ファイルをインポートする場合は、プロジェクト間の参照は ZIP ファイルに含まれておらず、Silk Test Workbench は新しく作成したプロジェクトに Common プロジェクトへの参照を追加します。

13. **次へ** をクリックして、インポートを開始します。 **インポートの進捗状況** ウィンドウに、完了レベルを示す進行状況バーが表示されます。インポートが完了すると、**インポートウィザードの完了** ページが開きます。

14. **省略可能** : **保存** をクリックして、詳細レポートを保存します。

15. **完了** をクリックします。

 **注:** Silk Test Workbench 18.5 以降を使用している場合、18.5 より前のバージョンの Silk Test Workbench でエクスポートした資産をインポートできません。以前のバージョンでエクスポートした資産をインポートするには、このような資産をインポートする場合は、

StwConvertExportedXML プログラムを使用してください。このコマンドラインプログラムを使用すると、資産を新しい形式に変換できます。このプログラムには、次の引数を指定できます。

STWConvertExportedXML [-overwrite] [-?] path_to_old_export path_to_new_export.zip

- path_to_old_export: Silk Test Workbench の以前のバージョンで生成したファイルまたはフォルダへのパス。
- path_to_new_export.zip: 変換したエクスポート ファイルを保存する ZIP ファイルへのパス。
- -overwrite: ZIP ファイルが既に存在するときに上書きする場合に指定します。-overwrite を指定しない場合、ZIP ファイルが存在すると、エラー メッセージを出力してプログラムが終了します。
- -?: ヘルプ メッセージを出力してプログラムが終了します。



注: 資産のインポートでは、履歴データはすべてインポート時の情報に置き換えられます。たとえば、作成者、作成日、最終更新者、更新日には、インポートを実行したユーザーおよびインポートの日付と時刻が反映されます。

資産のエクスポート

資産のデータベース間でのエクスポートは、**資産エクスポート** ウィザードを使用して行います。



注: エクスポート ファイルは変更しないでください。変更すると、ファイルのインポート時に Silk Test Workbench データベースが破損する可能性があります。資産の編集集中にエクスポートが発生した場合、編集集中の資産はエクスポートされません。そのため、エクスポートを実行する前に、資産に対するすべての編集を完了し、保存していることを確認してください。

1. **ファイル > 資産のエクスポート** を選択します。Silk Test Workbench **資産エクスポート** ウィザードが開きます。

このウィザードおよびそれ以外のすべての Silk Test Workbench ウィザードを今後使用するとき、**よろこ** ページをスキップするには、このページの最下部にあるチェックボックスをオンにします。このチェックボックスをオンにすると、**全般** オプションの **ウィザードの [よろこ] 画面の表示** オプションが更新されます。

ウィザードのサイズは、任意のページの右下隅をドラッグして変更できます。

2. **次へ** をクリックします。 **エクスポート対象** ページが開きます。
3. すべての資産をエクスポートするには、以下のステップを実行します。
 - a) **すべての資産をエクスポートする** をクリックします。
 - b) **各資産の最新バージョンだけをエクスポートする** チェックボックスをオフにして各資産のすべてのバージョンをエクスポートするか、このチェックボックスをオンにして最新バージョンだけをエクスポートに含めます。
4. 特定の資産をエクスポートするには、以下のステップを実行します。
 - a) **選択した資産だけをエクスポートする** をクリックします。
 - b) 各資産の特定のバージョンをエクスポートするには、**各資産のすべてのバージョンを表示する** チェックボックスをオンにします。



注: 資産の複数のバージョンをエクスポートするよう選択した場合は、選択するバージョンに応じて、バージョン情報がインポート時に変更することがあります。たとえば、資産のバージョン 1 と 5 をエクスポートするよう選択した場合、インポート時にバージョン番号が 1 と 2 に更新されます。

- c) **ビジュアル テストおよび .NET スクリプトに関連付けられた資産をエクスポートする** チェックボックスをオンにすると、すべての参照プロジェクト内のすべてのオブジェクト マップを含む、すべての関連資産をエクスポートできます。
- d) **次へ** をクリックします。 **エクスポートする項目** ページが開きます。
- e) 左側のペインでエクスポートする項目を含むフォルダを展開し、エクスポートする資産のチェックボックスをオンにします。

青いチェックマークは、項目とそのすべてのサブ項目がエクスポート対象として選択されていることを示します。灰色のチェックマークは、サブ項目の一部だけが選択されていることを示します。

5. **次へ** をクリックします。 **データの格納場所** ページが開きます。
6. 資産をエクスポートする ZIP ファイルの場所を指定します。
エクスポートしたすべての資産が、指定したファイルにまとめられます。
7. **次へ** をクリックします。 **資産のエクスポートの開始** ページが開きます。このページでは、エクスポートする資産の概要が表示されます。
8. 要約を確認し、**次へ** をクリックしてエクスポートを開始します。 **エクスポートの進捗状況** ウィンドウに、完了レベルを示す進行状況バーが表示されます。エクスポートが完了すると、**エクスポートウィザードの完了** ページが開きます。
9. 省略可能： **保存** をクリックして、詳細レポートを保存します。
- 10 **完了** をクリックします。

資産のインポートとエクスポートの権限

ユーザーに割り当てられている権限に応じて、資産のインポート機能を制限したり、無効にしたりすることができます。権限は、プロジェクトごとに固有に設定できます。たとえば、*ProjectA* に対する読み取り専用権限を持つユーザーに対して、*ProjectB* ではフル アクセス権限を割り当てることができます。この結果、このユーザーにとってインポート機能は、*ProjectA* で無効であっても、*ProjectB* では使用可能になります。

資産のエクスポート機能は、通常は制限されていません。読み取り専用以上の権限を持つすべてのユーザーは、プロジェクト資産とシステム資産の両方をエクスポートできます。

以下の表に、各ユーザー ロールが、プロジェクト資産およびシステム資産をインポートできるかどうかを示します。プロジェクト資産とシステム資産は、以下のように定義されます。

- プロジェクト資産：詳細については、「資産の種類」を参照してください。
- システム資産：記録オプションと再生オプション。

資産のインポート権限

管理者またはフル アクセス権限を持つユーザーのみが、すべてのプロジェクト資産とシステム資産をインポートできます。インポート権限の完全なリストについては、以下の表を参照してください。

ユーザー ロール	プロジェクト資産のインポート	システム資産のインポート
アクセス不可	いいえ	はい
読み取り専用	いいえ	はい
実行者	はい ¹	はい
スクリプト作成者	はい	はい
フル アクセス	はい	はい

¹：結果のインポートのみが許可されます。

資産のエクスポート権限

読み取り専用以上の権限を持つユーザーは、すべてのプロジェクト資産とシステム資産をエクスポートできます。

オブジェクト解決

Silk Test Workbench は、コントロール クラスの名前と優先付けられた属性のコレクションの組み合わせによって、テスト対象アプリケーション (AUT) のコントロールをユニークな XPath ロケーターに識別します。組み合わせた XPath ロケーターでコントロールをユニークに識別できない場合には、Silk Test Workbench は、ロケーターにインデックスを追加します。Silk Test Workbench で記録中に、**操作の選択** ダイアログの **ロケーター** フィールドのリストから、コントロールの他のロケーターを選択できます。

Silk Test Workbench では、識別されたオブジェクトのリテラル参照はロケーターと呼ばれます。Silk Test Workbench は、ロケーターを使用して、テスト対象アプリケーション (AUT) のオブジェクトを検索して識別します。ロケーターは、W3C (World Wide Web Consortium) によって定義された 共通の XML ベース言語である XPath クエリー言語のサブセットです。

ビジュアル テストまたはスクリプトを記録すると、Silk Test Workbench によってロケーターが自動的に記録されます。デフォルトでは、Silk Test Workbench は、コントロールのリテラル参照またはウィンドウのロケーター名を使用する代わりに、エイリアスまたは論理名 (オブジェクト マップ項目と呼ばれる) を使用します。ビジュアル テストおよびスクリプトでは、実際のロケーター名ではなく、オブジェクト マップ項目に基づいてオブジェクトが参照されます。

オブジェクト マップを無効にすると、オブジェクト マップ項目ではなくロケーターがビジュアル テストまたはスクリプト コンテキストに組み込まれます。

ロケーターの基本概念

Silk Test Workbench は、XPath クエリー言語のサブセットをサポートしています。XPath の詳細については、<http://www.w3.org/TR/xpath20/> を参照してください。

XPath 式は現在のコンテキスト、つまり、Find メソッドを呼び出したオブジェクトの階層上における位置に依存します。ファイル システムと同じように、すべての XPath 式は、この位置に依存します。例：

- `"//Shell"` は、現在のコンテキストから始まるすべての階層にあるすべての Shell を見つけます。
- `"Shell"` は、現在のコンテキストの直下の子であるすべての Shell を見つけます。

さらに、ある XPath 式は、コンテキストの影響を受けます。たとえば、`myWindow.find(xpath)` は、`myWindow` が現在のコンテキストとなります。

動的オブジェクト解決は、テスト ケース内でオブジェクトを識別するために、Find または FindAll メソッドを使用します。

オブジェクト タイプと検索範囲

典型的なロケーターには、検索するオブジェクトのタイプと検索範囲が含まれます。検索範囲は以下のいずれかです。

- `//`
- `/`

ロケーターは、ロケーターを指定する対象となるオブジェクトである、現在のオブジェクトに依存します。現在のオブジェクトは、アプリケーション UI のオブジェクト階層における位置を特定します。ファイル システムと同じように、すべてのロケーターは、この階層における現在のオブジェクトの位置に依存します。

XPath 式は、現在のコンテキスト、つまり、Find メソッドを呼び出したオブジェクトの階層上における位置に依存します。ファイル システムと同じように、すべての XPath 式は、この位置に依存します。

**注:**

HTML 要素に対するロケータにおけるオブジェクト タイプは、HTML タグ名または、このオブジェクトに対して Silk Test Workbench が使用するクラス名のいずれかになります。たとえば、ロケータ `//a` と `//DomLink` (ここで、`DomLink` は Silk Test Workbench でのハイパーリンクに対する名前です) は同じです。HTML ベースでないテクノロジーの場合は、Silk Test Workbench クラス名だけが使用されます。

例

- `//a` は、現在のオブジェクトに相対的なすべての階層にあるハイパーリンク オブジェクトを識別します。
- `/a` は、現在のオブジェクトの直下の子であるハイパーリンク オブジェクトを識別します。



注: `<a>` は、Web ページのハイパーリンクを表す HTML タグです。

例

以下のコード例は、ブラウザ内の最初のハイパーリンクを識別します。この例では、実行中のブラウザ インスタンスを参照するスクリプトに `browserWindow` という名前の変数が存在することを仮定しています。ここで、タイプは "a" で、現在のオブジェクトは `browserWindow` です。

```
Dim link As DomLink = browserWindow.DomLink("//a")
```

属性を使用したオブジェクトの識別

プロパティに基づいてオブジェクトを識別するために、ロケータ属性を使用できます。ロケータ属性は、オブジェクトの型の後にかぎ括弧で指定します。

例

次の例では、テキスト `Home` を持つハイパーリンクを識別するために、`textContent` 属性を使用します。同じテキストを持つハイパーリンクが複数存在する場合は、最初のハイパーリンクがロケータによって識別されます。

```
Dim link as DomLink = browserWindow.DomLink("//a[@textContent='Home']")
```

ロケータの構文

Silk Test Workbench では、UI コントロールを特定するために、XPath クエリ言語のサブセットをサポートしています。

以下の表に、Silk Test Workbench がサポートする構成子を一覧します。



注: `<a>` は、Web ページのハイパーリンクを表す HTML タグです。

サポートするロケータ構成子	例	説明
<code>//</code>	<code>//a</code>	現在のオブジェクトより下層にあるオブジェクトを識別します。

サポートするロケータ構成子	例	説明
/	/a	この例は、Web ページのハイパーリンクを識別します。 現在のオブジェクトの直下にあるオブジェクトを識別します。それより下層にあるオブジェクトは解決されません。
属性	例 1 : // a[@textContents='Home'] 例 2 : // div[@textContents='Price: * USD']	この例は、現在のオブジェクトの直下にある Web ページのハイパーリンクを識別します。 特定の属性によってオブジェクトを識別します。属性値には、ワイルドカード * および ? を使用できます。 例 1 は、テキスト <i>Home</i> を持つハイパーリンクを識別し、例 2 は、ワイルドカードを使って任意の価格を持つ div を識別します。
インデックス	例 1 : //a[3] 例 2 : // a[@textContents='Home'][2]	複数のオブジェクトがある場合に、インデックスを指定して特定のオブジェクトを識別します。ロケータのインデックスは 1 から始まります。 例 1 は、3 番目のハイパーリンクを識別し、例 2 は、テキスト <i>Home</i> を持つ 2 番目のハイパーリンクを識別します。
論理演算子 : and、or、not、=、!=	例 1 : // a[@textContents='Remove' or @textContents='Delete'] 例 2 : //a[@textContents! ='Remove'] 例 3 : // a[not(@textContents='Delete' or @id='lnkDelete') and @href='*/delete']	論理演算子を使って、属性を組み合わせてオブジェクトを識別します。 例 1 は、 <i>Remove</i> か <i>Delete</i> のテキストを持つハイパーリンクを識別し、例 2 は、 <i>Remove</i> 以外のテキストを持つハイパーリンクを識別し、例 3 は、複数の論理演算子を組み合わせた例を示しています。
ancestor	例 1 : //input[@id='username']/ ancestor::form 例 2 : //input[@id='username']/ ancestor::div[@className='c ontainer']	現在のオブジェクトより上層にあるオブジェクトを識別します。 例 1 は、識別子 <i>username</i> を持つ input 要素を子に持つ form 要素を識別し、例 2 は、識別子 <i>username</i> を持つ input 要素を子に持ち、クラス名が <i>container</i> を持つ div を識別します。
..	例 1 : // a[@textContents='Edit']/..	オブジェクトの親を識別します。 例 1 は、テキスト <i>Edit</i> を持つハイパーリンクの親を識別し、例 2 は、テキスト <i>Edit</i> を持つハイパーリンクと同

サポートするロケータ構成子	例	説明
	例 2 : // a[@textContents='Edit']/..// a[@textContents='Delete']	列にあるテスト <i>Delete</i> を持つハイパーリンクを識別します。
following-sibling	例 : // td[@textContents='John']/ following-sibling::td[2]	現在のオブジェクトより後ろにある同列オブジェクトを識別します。 この例は、テキスト <i>John</i> を持つテーブルセルの右側から 2 番目にあるテーブルセルを識別します。
preceding-sibling	例 : // td[@textContents='John']/ preceding-sibling::td[2]	現在のオブジェクトより前にある同列オブジェクトを識別します。 この例は、テキスト <i>John</i> を持つテーブルセルの左側から 2 番目にあるテーブルセルを識別します。
*	例 1 : // *[@textContents='Home'] 例 2 : /*/a	オブジェクトの型 (ハイパーリンク、テキストフィールド、ボタンなど) を考慮せずにオブジェクトを識別します。 例 1 は、オブジェクトの型は任意で、指定したテキストを持つオブジェクトを識別し、例 2 は、現在のオブジェクトより 2 つ下層にあるハイパーリンクを識別します。

以下の表には、Silk Test Workbench がサポートしないロケータ構成子を一覧します。

サポートしないロケータ構成子	例
右辺、左辺ともに属性を指定して比較する。	//a[@textContents = @id]
属性名を右辺に指定することはサポートされません。属性名は左辺に指定する必要があります。	//a['abc' = @id]
複数のロケータを and あるいは or で結合する。	//a[@id = 'abc'] or ../Checkbox
複数の属性をかぎ括弧で指定する。	//a[@id = 'abc'] [@textContents = '123'] (代わりに //a [@id = 'abc' and @textContents = '123'] を使用)
複数のインデックスをかぎ括弧で指定する。	//a[1][2]
クラスあるいはクラス名の一部にワイルドカードを含むクラスワイルドカードを明示的に指定しない構成子。	//[@id = 'abc'] (代わりに //*[@id = 'abc'] を使用) "/*//a[@id='abc']"

スクリプトでのロケータの使用

Silk Test Workbench では、識別されたオブジェクトのリテラル参照はロケータと呼ばれます。必要に応じて、ロケータ文字列の短縮形をスクリプトで使用できます。スクリプトを再生すると、Silk Test Workbench によって自動的に構文が展開されて完全なロケータ文字列が使用されます。スクリプトを手動でコーディングする場合は、次の順番で次の部分を省略できます。

- 検索スコープ「//」。
- オブジェクトの型名。Silk Test Workbench のデフォルトはクラス名です。
- 属性を囲む角かっこ「[]」。

スクリプトを手動で記述する場合は、使用可能な最も短い形式を使用することをお勧めします。



注: .NET スクリプトを使用する場合にのみ、ロケータ文字列の短縮形を使用できます。ビジュアルテストでは、完全なロケータ文字列を使用する必要があります。オブジェクトを識別する場合は、完全なロケータ文字列がデフォルトでキャプチャされます。

以下のロケータは同じです。

- 最初の例では、完全なロケータ文字列が使用されています。

```
_desktop.DomLink("//BrowserApplication//BrowserWindow//a[@textContents='Home']").Select()
```

完全なロケータ文字列を確認するには、**オブジェクトの識別** ダイアログ ボックスを使用します。

- 2 番目の例は、ブラウザー ウィンドウが既に存在する場合に機能します。

```
browserWindow.DomLink("//a[@textContents='Home']").Select()
```

または、短縮形を使用することができます。

```
browserWindow.DomLink("@textContents='Home']").Select()
```

識別のための実際の属性がないオブジェクトを検索するには、インデックスを使用します。たとえば、Web ページの 2 つめのハイパーリンクを選択するには、以下のように入力します。

```
browserWindow.DomLink("[2]").Select()
```

さらに、その種類の最初のオブジェクトを検索する（このことは、オブジェクトに実際の属性がない場合に便利です）には、以下のように入力します。

```
browserWindow.DomLink().Select()
```

Find メソッドの使用

.DomLink のようなメソッドを使用する代わりに、Find メソッドを使用してロケータを持つ単一のオブジェクトを識別することができます。



注: Find メソッドは、完全ロケータのみ使用できます。省略形のロケータはサポートされません。

以下のように入力する代わりに：

```
_desktop.DomLink("//a[@textContents='Home']").Select()
```

以下のように入力することができます。

```
_desktop.Find(Of DomLink)("//a[@textContents='Home']").Select()
```

.DomLink メソッドは、Find メソッドで内部的に使用されます。 .DomLink メソッドを使用した方が、Find メソッドよりも簡潔に表現できるため、好まれます。

ロケータを使用したオブジェクトの存在確認

Exists メソッドを使用して、オブジェクトがテスト対象アプリケーションに存在するかどうかを確認できます。

次のコードは、「Log out」というテキストのハイパーリンクが Web ページに存在するか確認します。

```
If (browserWindow.Exists( "//a[@textContents='Log out']" )) Then
    ' do something
End If
```

Find メソッドの使用

Find メソッドや FindOptions メソッドを使用して、後で使用したいオブジェクトが存在するか確認できます。

次のコードは、ウィンドウを検索し、ウィンドウが見つかった場合にウィンドウを閉じます。

```
Dim mainWindow As Window
mainWindow = _desktop.Find("//Window[@caption='My Window']", New FindOptions(False))
If (mainWindow IsNot Nothing) Then
    mainWindow.CloseSynchron()
End If
```

1 つのロケーターで複数のオブジェクトを識別する

FindAll メソッドを使用して、ロケーターに一致する最初のオブジェクトのみを識別するだけでなく、ロケーターに一致するすべてのオブジェクトを識別できます。

例

次のコードの例は、FindAll メソッドを使用して、Web ページのすべてのハイパーリンクを取得します。

```
Dim links As IList(Of DomLink) = browserWindow.FindAll(Of DomLink)
("//a")
```

ロケーターのカスタマイズ

このセクションでは、テスト対象アプリケーション (AUT) のコントロールを Silk Test Workbench が確実に解決できるようにするために、安定したロケーターを作成する方法について説明します。

Silk Test Workbench は、AUT がその UI コントロールに対して公開する識別子を利用して、非常に柔軟で強力な UI コントロールの識別方法を提供します。Silk Test Workbench は、任意の UI コントロールに対して宣言された任意のプロパティを使用して、UI コントロールの階層を使ってロケーターを作成できます。Silk Test Workbench は、それぞれの UI コントロールを識別するのに最も適した項目とプロパティを階層から選択します。

Silk Test Workbench は、UI コントロールの階層から多くのコントロールを動的に除外するため、AUT の変更に対して非常に影響を受けにくいオブジェクト解決方法を Silk Test Workbench は提供します。Web ページの書式要素のよな UI コントロール ツリーの階層を変更する中間のグループ化されたコントロールは、オブジェクト解決から除外することができます。

UI コントロールによっては、それを固有に識別できるようにする有意義なプロパティを公開しません。このようなコントロールを含んだアプリケーションは低いテスト容易性を持つアプリケーションとみなされます。階層、とくに動的な階層は、このようなアプリケーションに対する固有のロケーターを作成するのに、重要な意味を持ちます。高いテスト容易性を持つアプリケーションは、固有の UI コントロールを識別するための単純な仕組みを常に提供します。

AUT のテストを容易にする最も単純で最も効果的な慣例のひとつが、コントロールに対する安定した識別子を導入し、アプリケーションの既存のインターフェイスを通して、これらの安定した識別子を公開することです。

安定した識別子

UI コントロールの安定した識別子とは、コントロールの呼び出しごとに、または UI コントロールが存在するアプリケーションのバージョンが変わっても変更されない識別子を言います。安定した識別子は、その使用されるコンテキストにおいて一意である必要があります。つまり、同じ識別子を持つコントロール

が同時にアクセス可能でないことが求められます。つまり、グローバルなコンテキストで一意である GUID 形式の識別子を使用する必要はありません。コントロールの識別子は、可読性が高く、有意な名前であるべきです。これらの識別子の命名規則によって、実際のコントロールに識別子を関連付けるのがより容易になります。

例：キャプションはコントロールの良い識別子と言えるか

ほとんどのテストツールでは、UI コントロールのデフォルト識別子としてキャプションを使用します。キャプションは、コントロールに関連付けられた UI のテキストです。しかし、UI コントロールを識別するためにキャプションを使用することには、次のような欠点があります。

- キャプションは安定していません。キャプションは開発プロセスの間に頻繁に変更されます。たとえば、AUT の UI が開発プロセスの終わりにレビューされる場合があります。UI が安定していないため、開発プロセスの初期の段階で UI テストを導入することが困難になります。
- キャプションは一意ではありません。たとえば、アプリケーションには **OK** というキャプションを持つボタンが複数存在する可能性があります。
- 多くのコントロールはキャプションを表示しないため、識別するために、ほかのプロパティを使用する必要があります。
- ローカライズしたアプリケーションのテストにキャプションを使用する場合、各言語ごとにコントロールのキャプションを保守する必要があるため、非常に扱いにくく、さらに言語ごとに適切なキャプションを動的に割り当てることができるように、複雑なスクリプトロジックを保守する必要があります。

安定したロケータを作成する

Silk Test Workbench の主要なメリットのひとつが、柔軟で強力なオブジェクト解決の仕組みです。UI コントロールを特定するために XPath 記法を使用することによって、UI コントロールが適切な属性を持っていない場合でも、適切な属性を持つ対象要素のそばにある限り、Silk Test Workbench は確実に識別できます。Silk Test Workbench の XPath ロケータは、UI コントロールを識別するために、UI コントロールの階層全体を使用することもできれば、その一部を使用することもできます。特に最近の AJAX ツールキットは、とても複雑なドキュメントオブジェクトモデル (DOM) を動的に生成するので、UI コントロールを特定するために使用できる適切なコントロール属性を提供しません。

このような場合、インテリジェントなオブジェクト解決の仕組みを提供しないテストツールでは、UI コントロールを識別するために、ほとんどの場合インデックスベースの解決方法を使用することが必要になります。たとえば、展開アイコンの n 番目のコントロールを識別します。このようなテストスクリプトは保守することが容易でなく、アプリケーションにほんのわずかな変更を加えるだけでテストスクリプトが動作しなくなってしまうことが良くあります。

有用な属性を提供しない UI コントロールに対して安定したロケータを作成する良い方法は、階層の中で安定したロケータを持つアンカー要素を見つけることです。そして、そのアンカー要素からロケータを作成したい要素まで辿っていくことができます。

Silk Test Workbench は、この方法を使用してロケータを作成しますが、ときにはコントロールからの安定したロケータを手動で作成することが必要となる場合もあります。

例：コントロールの同列要素の指定

このトピックでは、ロケータに使用することのできる安定した属性を提供しないコントロールを、そのコントロールの同列要素に対する安定したロケータが利用できる場合に、それを使用して指定する方法を説明します。

次の安定したロケータを持つコントロール **Item 0.0** が既に識別されていると想定します。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='Item 0.0']
```

Item 0.0 が、タイプ *a* の後ろに同列要素を持っていることがわかっている場合、次のコードを使用して、同列要素の安定したロケータを構築できます。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='Item 0.0']/following-sibling::a
```

テキスト フィールドを識別するときにも、この同列要素による方法を使用できます。テキスト フィールドは、たいていの場合、ロケータで使用できる有用な属性を提供しません。テキスト フィールドのラベルを使用すると、テキスト フィールドの有用なロケータを作成できます。同列要素による方法を使って、テキスト フィールドのロケータの一部として、ラベルを使用することは簡単です。たとえば、テキスト フィールドがテキスト **User Name** を持つラベルの前に同列要素である場合、次のロケータを使用できます。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='User Name']/preceding-sibling::input[@type='text']
```

例：動的 GWT ツリーの展開アイコンの検索

Google Widget Toolkit (GWT) は、とても人気のある強力なツールキットですが、テストしにくいです。動的ツリー コントロールは、とても一般的に使用されている GWT の UI コントロールです。ツリーを展開するには、**展開** アイコン要素を識別する必要があります。

動的 GWT ツリーのサンプルは、<http://samples.gwtproject.org/samples/Showcase/Showcase.html#!CwTree> にあります。

Silk Test Workbench が生成するデフォルトのロケータは次のようになります。

```
/BrowserApplication//BrowserWindow//DIV[@id='gwt-debug-cwTree-dynamicTree-root-child0']/DIV/DIV[1]//IMG[@border='0']
```

次の理由で、このデフォルトのロケータは、**Item 0.0** の **展開** アイコンを識別するロケータとしては信頼できるものではありません。

- ロケータは複雑で、複数の階層から構成されています。AJAX で動的に DOM 構造が少し変わるとロケータは使えなくなります。
- ロケータには、階層のいくつかのコントロールにインデックスが含まれています。インデックスベースのロケータは、その出現番号によってコントロールを検索するため、一般にもろく、たとえば、ツリーの 6 番目の展開アイコンを見つけるなど、うまく特定のコントロールを定義できません。このルールの例外は、たとえばグリッドの 6 番目のデータ行など、識別するさまざまなデータ セットを表すためにインデックスが使用される場合です。

多くの場合、より良いロケータを見つける良い方法は、検索する要素の同列要素を探し出すことです。より良いロケータの同列要素を見つけると、XPath は、これらの同列要素を識別してロケータを構成することができます。この場合、ツリー項目 **Item 0.0** は、**展開** アイコンよりも良いロケータです。ツリー項目 **Item 0.0** のロケータは、コントロールの @textContents プロパティを使用するため、安定した単純なロケータです。

デフォルトでは、Silk Test Workbench は @id プロパティを使用しますが、GWT では @id には 'gwt-uid-<nnn>' のような値 (ここで、<nnn> は同じ要素でも呼出し毎に頻繁に変わります) が含まれるため、たいてい安定したプロパティではありません。

@textContents プロパティを @id の代わりに使用してロケータを手動で変更できます。

元のロケータ：

```
/BrowserApplication//BrowserWindow//DIV[@id='gwt-uid-109']
```

別のロケータ：

```
/BrowserApplication//BrowserWindow//DIV[@textContents='Item 0.0']
```

もしくは、@id='gwt-uid-<nnn>' を使用しないように Silk Test Workbench を設定できます。この場合、Silk Test Workbench は自動的に安定したロケータを記録します。たとえば、@id プロパティで使われるテキスト パターンをロケータ属性値除外リストに追加します。この場合、gwt-uid* を除外リストに追加します。

要素の階層を調べると、**Item 0.0** コントロールと **展開** アイコン コントロールは、共通のルートノードとして DomTableRow コントロールを持つことが分かります。

展開 アイコンの安定したロケータを作成するには、次のロケータを使って **Item 0.0** をまず検索する必要があります。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='Item 0.0']
```

そして、要素の階層を 2 レベル上がって DomTableRow 要素まで移動します。これは、XPath では、ロケータに `../../` を追加して表現します。最後に、DomTableRow から **展開** アイコンを検索します。**展開** アイコンは、サブツリー内では唯一の IMG コントロールであるので、容易に検索できます。これは、XPath では、ロケータに `//IMG` を追加して表現します。**展開** アイコンの最終的な安定したロケータは次のようになります。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='Item 0.0']/../../IMG
```

XPath の ancestor 軸を使うと、より簡単に **展開** アイコンを指定できます。

```
/BrowserApplication//BrowserWindow//DIV[@textContent='Item 0.0']/ancestor::tr//IMG
```

カスタム属性

多くの UI テクノロジーは、UI コントロールのあらかじめ定義された属性のセットをカスタム属性で拡張する方法を提供します。アプリケーション開発者は、コントロールを一意に識別する安定した識別子を導入するためにカスタム属性を使用できます。Silk Test Workbench は、UI コントロールのカスタム属性にアクセスでき、UI コントロールを識別するために、これらのカスタム属性を使用することもできます。

UI コントロールを識別するために特別に自動化用属性を使用すると、caption のような定義済み属性を使用する場合と比較して、いくつかのメリットを享受できます。アプリケーション コードで安定した識別子を指定でき、カスタム属性や定義済みの自動化用プロパティの何れかを通して識別子を公開することで、テスト自動化スクリプトが理解しやすくなり、保守性も高まり、開発プロセスの初期の段階からテストの自動化を開始することができるようになります。

Silk Test Workbench はロケータ生成の柔軟性が高く、識別に使用する属性を設定することができます。


Apache Flex アプリケーションのカスタム属性

Apache Flex アプリケーションは、あらかじめ定義されたプロパティ automationName を使用して、次のように Apache Flex コントロールに対して安定した識別子を指定します。

```
<?xml version="1.0" encoding="utf-8"?>
  <s:Group xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:mx="library://ns.adobe.com/flex/mx" width="400" height="300">
    <fx:Script>
      ...
    </fx:Script>
    <s:Button x="247" y="81" label="Button" id="button1" enabled="true"
      click="button1_clickHandler(event)"
      automationName="AID_buttonRepeat"/>
    <s:Label x="128" y="123" width="315" height="18" id="label1" verticalAlign="middle"
      text="awaiting your click" textAlign="center"/>
  </s:Group>
```

Apache Flex アプリケーションのロケータは次のようになります。

```
...//SparkApplication//SparkButton[@caption='AID_buttonRepeat']
```

 **注目:** Apache Flex アプリケーションの場合、Silk Test Workbench では automationName はロケータ属性 caption に常にマップされます。automationName 属性が指定されていない場合、Silk Test Workbench は属性 ID をロケータ属性 caption にマップします。

Java SWT カスタム属性

カスタム属性をテスト アプリケーションに追加して、テストをより安定させることができます。たとえば、Java SWT では、GUI を実装する開発者が属性 ('silkTestAutomationId' など) をウィジェットに対して定義することによって、アプリケーション内でそのウィジェットを一意に識別することができます。これにより、Silk Test Workbench を使用するテスト担当者は、その属性 (この場合は 'silkTestAutomationId') をカスタム属性のリストに追加すると、その一意の ID によってコントロールを識別できるようになります。カスタム属性を使用すると、caption や index のような他の属性よりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は定義済みのウィジェットより前に他のウィジェットが追加されると変更されるためです。

複数のオブジェクトに同じカスタム属性の値が割り当てられた場合は、そのカスタム属性を呼び出したときにその値を持つすべてのオブジェクトが返されます。たとえば、一意の ID として 'loginName' を 2 つの異なるテキスト フィールドに割り当てた場合は、'loginName' 属性を呼び出したときに、両方のフィールドが返されます。

Java SWT の例

以下のコードを使用して、テストするアプリケーションにボタンを作成する場合：

```
Button myButton = Button(parent, SWT.NONE);  
myButton.setData("SilkTestAutomationId", "myButtonId");
```

テストの XPath クエリ文字列に属性を追加するには、以下のクエリを使用します。

```
Dim button =  
desktop.PushButton("@SilkTestAutomationId='myButton'")
```

Java SWT アプリケーションをカスタム属性のテストに対して有効化にするには、開発者はカスタム属性をアプリケーションに含める必要があります。属性を含めるには org.swt.widgets.Widget.setData(String key, Object value) メソッドを使用します。

Web アプリケーションのカスタム属性

HTML は、安定した識別子を表すことができる一般的な属性 ID を定義します。定義により、ID は文書内の要素を一意に識別します。特定の ID を持つ要素は文書内で 1 つだけ存在します。

ただし、多くの場合 (特に AJAX アプリケーションでは)、ID は HTML 要素に関連付けられたサーバー ハンドラを動的に識別するために使用されます。つまり、Web 文書の作成のたびに ID は変わるようになります。このような場合、ID は安定した識別子ではなく、Web アプリケーションの UI コントロールを識別するのに適しません。

Web アプリケーションの場合、より確実にするには、Silk Test Workbench に UI コントロールの情報を公開するためだけに使用されるカスタム HTML 属性を新たに導入することです。

カスタム HTML 属性はブラウザーは無視するため、AUT の動作は変わりません。ブラウザーの DOM を通じてアクセスすることができます。Silk Test Workbench では、このような属性を (属性がコントロール クラスのカスタム 属性であっても) 識別時のデフォルト属性として使用するように設定することができます。特定のテクノロジ ドメインのデフォルト識別属性としてカスタム属性を設定するには、**ツール > オプション > 記録** をクリックして、テクノロジ ドメインを選択します。

アプリケーション開発者は、Web 要素にさらに HTML 属性を追加することが必要です。

元の HTML コード：

```
<A HREF="http://abc.com/control=4543772788784322..." <IMG  
src="http://abc.com/xxx.gif" width=16 height=16> </A>
```

新しいカスタム HTML 属性 *AUTOMATION_ID* を持つ HTML コード :

```
<A HREF="http://abc.com/control=4543772788784322..."  
AUTOMATION_ID = "AID_Login" <IMG src="http://abc.com/xxx.gif"  
width=16 height=16> </A>
```

カスタム属性を設定すると、Silk Test Workbench は、できる限りカスタム属性を使用して、一意のロケータを構成しようとしてします。Web ロケータは次のようになります。

```
...//DomLink[@AUTOMATION_ID='AID_Login']
```

例 : 変化する ID

変化する ID の 1 例は、Google Widget Toolkit (GWT) で、ID は Web 文書の作成のたびに変化する動的な値を保持します :

```
ID = 'gwt-uid-<nnn>'
```

この場合、<nnn> が頻繁に変化します。

Windows Forms アプリケーションのカスタム属性

Windows Forms アプリケーションは、あらかじめ定義された自動化用プロパティ *automationId* を使用して、Windows Forms コントロールに対して安定した識別子を指定します。

Silk Test Workbench は、ロケータを識別するために、自動的にこのプロパティを使用します。Windows Forms アプリケーションのロケータは次のようになります。

```
/FormsWindow//PushButton[@automationId='btnBasicControls']
```

WPF アプリケーションのカスタム属性

WPF アプリケーションは、あらかじめ定義された自動化用プロパティ *AutomationProperties.AutomationId* を使用して、次のように WPF コントロールに対して安定した識別子を指定します。

```
<Window x:Class="Test.MainWindow"  
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
Title="MainWindow" Height="350" Width="525">  
<Grid>  
  <Button AutomationProperties.AutomationId="AID_buttonA">The  
Button</Button>  
</Grid>  
</Window>
```

Silk Test Workbench は、ロケータを識別するために、自動的にこのプロパティを使用します。WPF アプリケーションのロケータは次のようになります。

```
/WPFWindow[@caption='MainWindow']//WPFButton[@automationId='AID_buttonA']
```

XPath のパフォーマンス問題のトラブルシューティング

複雑な Web アプリケーションの場合など、オブジェクトの構造が複雑なアプリケーションをテストする場合、パフォーマンスの問題や、スクリプトの信頼性に関連する問題が発生することがあります。このトピックでは、Silk Test Workbench が記録中に自動的に生成したロケータとは異なるロケータを使用して、スクリプトのパフォーマンスを向上させる方法について説明します。



注: 一般に、複雑なロケータを使用することは推奨しません。複雑なロケータを使用すると、テストの信頼性を損なう恐れがあります。複雑なロケータは、テスト アプリケーションの構造をほんの少し変更しただけで機能しなくなってしまう可能性があります。それにもかかわらず、スクリプト

のパフォーマンスが要求を満たしていない場合には、より固有のロケータを使用することによってテストのパフォーマンスを向上できる可能性があります。

例として、MyApplication アプリケーションの要素ツリーを以下に示します。

```
Root
  Node id=1
    Leaf id=2
    Leaf id=3
    Leaf id=4
    Leaf id=5
  Node id=6
    Node id=7
      Leaf id=8
      Leaf id=9
    Node id=9
      Leaf id=10
```

以下の最適化手法のいくつかを使用して、スクリプトのパフォーマンスを改善させることができます。

- 複雑なオブジェクト構造内の要素を特定したい場合は、オブジェクト構造全体ではなく、その特定の部分だけを検索するようにします。たとえば、サンプルツリーの識別子 7 を持つ要素を検索する場合に `Root.Find("//Node[@id='7']")` というクエリを使用している場合、`Root.Find("//Node[@id='6']/Node[@id='7']")` というクエリで置き換えます。最初のクエリは、識別子が 1 から 7 の要素に対して要素ツリーが検索されます。2 番目のクエリでは、識別子 1 のノードと識別子 6 のノードがある最初のレベルのノードにある要素がまず検索された後で、識別子 6 のノードのサブツリー内から識別子 7 の最初のリーフが検索されます。
- 同じ階層内にある複数の項目を特定したい場合は、まずは階層を特定してから共通のルートノードを利用して項目を特定します。`Root.FindAll("//Node[@id='1']/Leaf")` というクエリを使用している場合、次のようなクエリで置き換えます。

```
Public Sub Main()
  Dim commonRootNode = _desktop.Find("//Node[@id='1']")
  commonRootNode.Find("/Leaf[@id='2']")
  commonRootNode.Find("/Leaf[@id='3']")
  commonRootNode.Find("/Leaf[@id='4']")
  commonRootNode.Find("/Leaf[@id='5']")
End Sub
```



Silk Test Open Agent

Silk Test Open Agent は、スクリプトのコマンドを GUI 固有のコマンドに翻訳するソフトウェア プロセスです。つまり、Open Agent がテストするアプリケーションを動かす、監視しています。

ホストマシン上で 1 つのエージェントをローカルに実行できます。ネットワーク環境では、任意の数のエージェントがリモート マシン上でテストを再生できます。ただし、記録はローカル マシン上でのみ実行できます。

Silk Test Open Agent の起動

テストの作成またはサンプル スクリプトの実行前に、Silk Test Open Agent が実行されている必要があります。通常は、製品を起動したときにエージェントが実行されます。Open Agent を手動で開始しなければならない場合には、次のステップを実行してください。

(Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > ツール > Silk Test Open Agent**、または (Microsoft Windows 10) **スタート > Silk > Silk Test Open Agent** をクリックします。Silk Test Open Agent アイコン  が、システムトレイに表示されます。


テスト実行後に Open Agent を停止する

コマンドライン オプション `-shutDown` を使うか、スクリプトから Open Agent を停止して、テスト実行が終わってもエージェントが実行し続けることがないようにすることができます。次のコマンドをコマンドラインから実行して、Open Agent を停止することができます：

```
openAgent.exe -shutDown
```


スクリプトからエージェントを停止するには：

1. テスト実行が完了したときに実行するスクリプトを開く、または作成します。
たとえば、テスト実行後にクリーンアップのために使用する既存のスクリプトを開きます。
2. スクリプトに ShutDown メソッドを追加します。

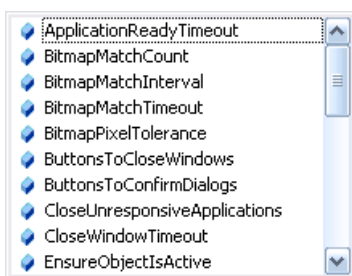
 **注:** 他のスクリプトが Open Agent を必要になった時点で、エージェントは再起動されます。

エージェント オプション

このセクションでは、GetOption および SetOption メソッドで操作できるオプションについて説明します。

 **注:** デフォルトでは、メニューから **ツール > オプション** を選択してグローバル オプションを設定できます。

スクリプトでエージェント オプションを設定または取得するには、GetOption または SetOption メソッドを使用します。たとえば、`Agent.SetOption(Options. と入力すると、オートコンプリートと構文ヒントテクノロジーを使用して、エージェント オプションがアクティブなエディター ウィンドウに自動的に表示されます。たとえば、以下のオプションがアクティブなエディター ウィンドウに表示されます。`



スクリプトでオプションを挿入する場合は、次のように記述します。

```
'VB .NET code
Agent.SetOption(Options.ApplicationReadyTimeout, 100000)
```


あるいは、スクリプトでエージェント オプション名を使用することもできます。たとえば、以下のように入力できます。

```
'VB .NET code
Agent.SetOption("OPT_APPREADY_TIMEOUT", 100000)
```


Open Agent の設定 を使用して、ビジュアル テストにカスタム オプションを追加することができます。**ツール > オプション** に移動し、設定するオプションの種類に依存して **記録** または **再生** を選択し、新しいオプションを名前値ペアとして追加します。この機能は特定の場合にのみ使用すべきです（サポートから提案された場合など）。

.NET オプション名	エージェント オプション名	定数の型	説明
ApplicationReadyTimeout	OPT_APPREADY_TIMEOUT	数値	新しく起動したアプリケーションが準備完了状態になるまで待機する時間をミリ秒で指定します。指定したタイムアウト時間内にアプリケーションの準備が整わない場合は、Silk Test Workbench によって例外がスローされます。
BitmapMatchCount	OPT_BITMAP_MATCH_COUNT	整数	ビットマップが安定していると判断するために、連続して同じビットマップでなければならないスナップショットの数を指定します。スナップショットは、OPT_BITMAP_MATCH_TIMEOUT で指定された秒数まで取得され、各スナップショット間で OPT_BITMAP_MATCH_INTERVAL で指定された時間間隔で一時的に停止します。デフォルトで、これは 0 です。
BitmapMatchInterval	OPT_BITMAP_MATCH_INTERVAL	実数	ビットマップ画像が安定していることを保証するために使用するスナップショット間の時間間隔を指定します。スナップショットは、OPT_BITMAP_MATCH_TIMEOUT で指定された時間まで取得されます。デフォルトで、これは 0.1 です。
BitmapMatchTimeout	OPT_BITMAP_MATCH_TIMEOUT	実数	ビットマップ画像が安定するまでの合計許容時間を指定します。

.NET オプション名	エージェント オプション名	定数の型	説明
			<p>この期間に Silk Test Workbench は画像の複数のスナップショットを取得し、スナップショット間で OPT_BITMAP_MATCH_TIMEOUT で指定された秒数だけ待機します。OPT_BITMAP_MATCH_COUNT で指定されたビットマップ数と一致する前に、OPT_BITMAP_MATCH_TIMEOUT から返される値に到達すると、Silk Test Workbench でスナップショットの取得が停止し、例外 E_BITMAP_NOT_STABLE が発生します。</p> <p>デフォルトで、これは 5 です。</p>
BitmapPixelTolerance	OPT_BITMAP_PIXEL_TOLERANCE	整数	<p>2 つのビットマップが一致しているとみなされる許容差異 (ピクセル単位) を指定します。差異ピクセル数がこのオプションで指定された値より小さい場合、ビットマップは同一であるとみなされます。最大許容値は 32767 ピクセルです。</p> <p>デフォルトで、これは 0 です。</p>
ButtonsToCloseWindows	OPT_CLOSE_WINDOW_BUTTONS	文字列のリスト	CloseSynchron メソッドでウィンドウを閉じるために使用するボタンを指定します。
ButtonsToConfirmDialogs	OPT_CLOSE_CONFIRM_BUTTONS	文字列のリスト	CloseSynchron メソッドでウィンドウを閉じる際に表示された確認ダイアログ ボックスを閉じるために使用するボタンを指定します。
CloseUnresponsiveApplications	OPT_KILL_HANGING_APPS	ブール値	無応答のアプリケーションを閉じるかどうかを指定します。タイムアウトなどの理由で、エージェントとアプリケーション間の通信に失敗した場合に、アプリケーションが無応答になります。複数のインスタンスを実行できないアプリケーションをテストする場合は、このオプションを TRUE に設定します。デフォルトで、これは FALSE です。
CloseWindowTimeout	OPT_CLOSE_WINDOW_TIMEOUT	数値	ウィンドウ閉じるための次の方式を試行する前に待機する時間をミリ秒で指定します。最終的に失敗と判断する前に、4 種類の方式が Agent によって実行されます。つまり、閉じるのに失敗するまでにかかる合計時間は、指定した値の 4 倍の時間になります。
Compatibility	OPT_COMPATIBILITY	文字列	Silk Test Workbench の最新バージョンで特定の機能の動作が変更されている場合は、これらの機能について、指定

.NET オプション名	エージェント オプション名	定数の型	説明
			<p>した Silk Test Workbench バージョンの動作を使用することができます。</p> <p>文字列の例：</p> <ul style="list-style-type: none"> • 12 • 11.1 • 13.0.1 <p>デフォルトでは、このオプションは設定されていません。</p>
EnsureObjectIsActive	OPT_ENSURE_ACTIVE_OBJDEF	ブール値	<p>ターゲット オブジェクトがアクティブであることを保証します。デフォルトで、これは FALSE です。</p>
EnableAccessibility	OPT_ENABLE_ACCESSIBILITY	ブール値	<p>Win32 アプリケーションをテストしているときに、Silk Test Workbench がオブジェクトを認識できない場合、ユーザー補助を有効にするには TRUE に設定します。ユーザー補助は、オブジェクトの認識機能をクラス レベルで強化するためのものです。</p> <p>ユーザー補助を無効にするには FALSE に設定します。</p> <p>デフォルトで、これは FALSE です。</p> <p> 注: Mozilla Firefox と Google Chrome の場合、ユーザー補助は常に有効で、無効にすることはできません。</p>
EnableMobileWebViewFallbackSupport	OPT_ENABLE_MOBILE_WEBVIEW_FALLBACK_SUPPORT	ブール値	<p>デフォルトのブラウザ サポートではテストできないハイブリッド モバイルアプリケーションに対して、モバイルネイティブ フォールバックのサポートを有効化します。</p> <p>デフォルトで、これは FALSE です。</p>
EnableUiAutomationSupport	OPT_ENABLE_UI_AUTOMATION_SUPPORT	数値	<p>通常の Win32 コントロールの認識の代わりに Microsoft UI オートメーションサポートを有効化する場合は TRUE を設定します。Win32 アプリケーションをテストしているときに、Silk Test Workbench がオブジェクトを認識できない場合、このオプションは有効です。AUTODETECT を指定すると、JavaFX などの既存のテクノロジーに対して Microsoft UI オートメーションが自動的に有効になります。</p> <p>デフォルトで、これは FALSE です。</p>
HangAppTimeout	OPT_HANG_APP_TIME_OUT	数値	<p>応答のない再生操作をキャンセルするまでのタイムアウト値を、無応答のアプリ</p>

.NET オプション名	エージェント オプション名	定数の型	説明
HighlightObjectDuringPlayback	OPT_REPLAY_HIGHLIGHT	ブール値	<p>リケーションのタイムアウトで指定します。</p> <p>デフォルト値は、5000 ミリ秒です。</p> <p>現在のオブジェクトが再生中にハイライトされるかどうかを指定します。</p> <p>デフォルトで、これは FALSE です。つまり、デフォルトではオブジェクトはハイライトされません。</p>
KeyboardEventDelay	OPT_KEYBOARD_INPUT_DELAY	数値	<p>再生時のキー入力間の遅延をミリ秒で指定します。</p> <p>テストするアプリケーションに応じて、選択する最適な値が異なることに注意してください。たとえば、Web アプリケーションをテストする場合、1 ミリ秒に設定すると、ブラウザが極端に遅くなります。ただし、この値を 0 (ゼロ) に設定すると、基本的なアプリケーションテストに失敗する可能性があります。</p>
KeysToCloseDialogs	OPT_CLOSE_DIALOG_KEYS	文字列のリスト	<p>CloseSynchron メソッドでウィンドウを閉じる際に表示されたダイアログボックスを閉じるキー シーケンスを指定します。例 : 「<ESC>」、「<Alt +F4>」</p>
LocatorAttributesCaseSensitive	OPT_LOCATOR_ATTRIBUTES_CASE_SENSITIVE	ブール値	<p>はいに設定すると、ロケータ属性名の大文字と小文字が区別されるようになります。いいえに設定すると、ロケータ属性名の大文字と小文字が区別されなくなります。モバイル Web アプリケーションのロケータ属性の名前は、常に大文字と小文字の区別はされません。つまり、モバイル Web アプリケーションの記録や再生時に、このオプションは無視されます。</p>
MenuItemsToCloseWindows	OPT_CLOSE_WINDOW_MENUS	文字列のリスト	<p>CloseSynchron メソッドでウィンドウを閉じるために使用するメニュー項目を指定します。例 : 「ファイル/*終了」、「ファイル/*中止」</p>
MouseEventDelay	OPT_MOUSE_INPUT_DELAY	数値	<p>各マウス イベントの前に使用される遅延をミリ秒で指定します。</p>
ObjectResolveRetryInterval	OPT_WAIT_RESOLVE_OBJDEF_RETRY	数値	<p>再生中にオブジェクトが解決できなかった場合に、再試行する前に待機する時間をミリ秒で指定します。</p> <p>ObjectResolveTimeout の値を超過した場合は、再試行は行われません。</p>
ObjectResolveTimeout	OPT_WAIT_RESOLVE_OBJDEF	数値	<p>再生中にオブジェクトの解決を待機する時間をミリ秒で指定します。オブジェクトが解決されるとすぐに、Silk Test</p>

.NET オプション名	エージェント オプション名	定数の型	説明
PlaybackMode	OPT_REPLAY_MODE	数値	<p>Workbench はこのオブジェクトを認識することができます。</p> <p>コントロールの再生方法を定義します。「低レベル」を使用すると、マウスとキーボードを使用して各コントロールが再生されます。「高レベル」を使用すると、API を使用して各コントロールが再生されます。コントロールごとにデフォルトの再生モードが割り当てられています。デフォルトの再生モードが選択されると、各コントロールごとのデフォルトの再生モードが使用されます。デフォルトモードを使用すると、最も信頼できる結果が得られます。「低レベル」または「高レベル」の再生を選択すると、すべてのコントロールの再生モードが選択した再生モードで上書きされます。</p> <p>有効な値は、0、1、2 です。0 はデフォルト、1 は高レベル、2 は低レベルです。デフォルトで、これは 0 です。</p>
PostReplayDelay	OPT_POST_REPLAY_DELAY	数値	<p>関数の呼び出しまたはプロパティの設定後に待機する時間をミリ秒で指定します。</p>
RemoveFocusOnCaptureText	OPT_REMOVE_FOCUS_ON_CAPTURE_TEXT	ブール値	<p>はいに設定すると、テキストのキャプチャ中にテスト対象アプリケーションからフォーカスが外れます。デフォルトではいいえに設定されており、テスト対象アプリケーションにフォーカスが残ります。テキストのキャプチャは、次のメソッドによる記録および再生中に実行されます。</p> <ul style="list-style-type: none"> • TextClick • TextCapture • TextExists • TextRect
SuspensionTimeout	OPT_NTF_SUSPENSION_TIMEOUT	数値	<p>NTF から回答を得るまで待機する最大時間をミリ秒で指定します。この時間を超えた場合は、NTF と Open Agent 間の通信が中断されたものとみなします。</p>
SyncTimeout	OPT_SYNC_TIMEOUT	数値	<p>オブジェクトが準備完了状態になる最大時間をミリ秒で指定します。</p> <p> 注: Silk Test 13.0 より前のバージョンの Silk Test からアップグレードする場合、OPT_XBROWSER_SYNC_TIMEOUT オプションが設定されていると、オブ</p>

.NET オプション名	エージェント オプション名	定数の型	説明
			<p>シオン ダイアログ ボックスに OPT_SYNC_TIMEOUT のデフォルト値が表示されますが、タイムアウトは定義した値に設定されています。</p>
TransparentClasses	OPT_TRANSPARENT_CLASSES	文字列のリスト	<p>オブジェクト階層を単純化し、テストスクリプトや関数のコードの行の長さを短くするために、次のテクノロジーの確実に不要なクラスに対するコントロールを抑制できます。</p> <ul style="list-style-type: none"> • Win32 • Java AWT/Swing • Java SWT/Eclipse • Windows Presentation Foundation (WPF) <p>記録や再生中に無視したいクラスの名前を指定します。</p>
	OPT_WPF_CHECK_DISPATCHER_FOR_IDLE	ブール値	<p>WPF アプリケーションによっては、コントロールの実装方法が影響して、Silk Test 同期が機能しない場合があります。これにより、WPF アプリケーションがアイドルであることを Silk Test Workbench が認識できなくなります。このオプションを FALSE に設定すると、WPF 同期が無効になり、WPF アプリケーションを制御するスレッドである WPF ディスパッチャーを Silk Test Workbench が確認しないようになります。このオプションを FALSE に設定すると、一部の WPF アプリケーションで発生する同期問題を解決できます。デフォルトで、これは TRUE です。</p>
WPFCustomClasses	OPT_WPF_CUSTOM_CLASSES	文字列のリスト	<p>記録や再生の対象にしたい WPF クラスの名前を指定します。たとえば、MyGrid というカスタム クラスが WPF Grid クラスから継承された場合、MyGrid カスタム クラスのオブジェクトは記録や再生に使用できません。Grid クラスはレイアウト目的のためのみ存在し、機能テストとは無関係であるため、Grid オブジェクトは記録や再生に使用できません。この結果、Grid オブジェクトはデフォルトでは公開されません。機能テストに無関係なクラスに基づいたカスタム クラスを使用するには、カスタム クラス (この場合は MyGrid) を</p> <p>OPT_WPF_CUSTOM_CLASSES オプションに追加します。これによって、記録、再生、検索、プロパティの検証な</p>

.NET オプション名	エージェント オプション名	定数の型	説明
WPFPrefillItems	OPT_WPF_PREFILL_ITEMS	ブール値	ど、すべてのサポートされる操作を指定したクラスに対して実行できるようになります。
XbrowserEnableIframeSupport	OPT_XBROWSER_ENABLE_IFRAME_SUPPORT	ブール値	記録および再生中に、WPFComboBox や WPFListBox のような WPFItemsControl 内の項目を事前に入力するかどうかを定義します。WPF 自体が特定のコントロールの項目を遅延読み込みするため、項目がビューにスクロールされない場合、それらの項目は Silk Test Workbench では使用できません。ビューにスクロールされないとアクセスできない項目にアクセスするには、事前入力をオンにします。これはデフォルトの設定です。ただし、一部のアプリケーションでは Silk Test Workbench によってバックグラウンドで項目が事前入力されると問題が発生し、そのためアプリケーションがクラッシュすることがあります。この場合、事前入力をオフにします。
XBrowserExcludeIFrames	OPT_XBROWSER_EXCLUDE_IFRAMES	文字列	ブラウザの iframe および frame のサポートを有効にするかどうかを指定します。Web アプリケーションの iframe の内容を無視して良い場合、iframe のサポートを無効にすると再生速度が速くなる可能性があります。たとえば、多くの広告が表示される Web ページやモバイル ブラウザーでテストする際に iframe サポートを無効にすると、再生パフォーマンスが大幅に向上する可能性があります。このオプションは、Internet Explorer では無視されません。このオプションは、デフォルトで有効になっています。
XBrowserFindHiddenInputFields	OPT_XBROWSER_FIND_HIDDEN_INPUT_FIELDS	ブール値	リストの各項目で、属性名と対応する値を定義します。項目のどれにも一致しない iframe と frame は、テスト対象に含まれます。ワイルドカードを使用できます。たとえば、"src:*advertising*" という項目を定義すると、<IFRAME src=http://my.domain/advertising-banner.html> がテスト対象から除外されます。このオプションは、Internet Explorer では無視されます。リストが空の場合は、すべての iframe と frame がテストの対象となります。複数のエントリをカンマで区切って指定します。

.NET オプション名	エージェント オプション名	定数の型	説明
XBrowserIncludeIFrames	OPT_XBROWSER_INCLUDE_IFRAMES	文字列	<p>type="hidden" を指定した HTML フィールドです)。デフォルト値は、TRUE です。</p> <p>リストの各項目で、属性名と対応する値を定義します。項目のどれにも一致しない iframe と frame は、テスト対象から除外されます。ワイルドカードを使用できます。たとえば、"name:*form" という項目を定義すると、<IFRAME name="user-form" src=...> がテスト対象に含まれます。このオプションは、Internet Explorer では無視されません。リストが空の場合は、すべての iframe と frame がテストの対象となります。複数のエントリをカンマで区切って指定します。</p>
XBrowserSynchronizationMode	OPT_XBROWSER_SYNC_MODE	文字列	<p>サポートする同期モード (HTML または AJAX) を設定します。HTML モードを使用すると、すべての HTML ドキュメントが対話的な状態になることが保証されます。このモードでは、単純な Web ページをテストすることができます。Java Script が含まれる、より複雑なシナリオが使用される場合は、同期関数を手動で記述することが必要になる可能性があります。AJAX モードを使用すると、同期関数を手動で記述する必要はなくなります。デフォルト値は、AJAX に設定されています。</p>
XBrowserSynchronizationTimeout	OPT_XBROWSER_SYNC_TIMEOUT	数値	<p>オブジェクトが準備完了状態になる最大時間をミリ秒で指定します。</p> <p> 注: 推奨されません。代わりに OPT_SYNC_TIMEOUT オプションを使用してください。</p>
XBrowserSynchronizationURLExcludes	OPT_XBROWSER_SYNC_EXCLUDE_URLS	文字列	<p>ページ同期中に除外するサービスまたは Web ページの URL を指定します。AJAX フレームワークやブラウザによっては、サーバーから非同期にデータを取得するために、特殊な HTTP 要求を継続して出し続けるものがあります。これらの要求により、指定した同期タイムアウトの期限が切れるまで同期がハングすることがあります。この状態を回避するには、HTML 同期モードを使用するか、問題が発生する要求の URL を 同期除外リスト 設定で指定します。</p> <p><i>http://test.com/timeService</i> のように URL 全体を入力したり、<i>timeService</i> のように URL の一部を入力します。</p>

.NET オプション名	エージェント オプション名	定数の型	説明
			<p>次のようにカンマで項目を区切ります。</p> <pre>'VB .NET code Agent.SetOption(OPT_XBROWS ER_SYNC_EXCLUDE_URLS, { "fpdownload.macromedia.co m", "fpdownload.adobe.com", "download.microsoft.com" })</pre>

Silk Test Workbench コンポーネント間の接続設定

ファイアウォールを介してリモートマシンに接続したり、HTTPS を使って安全にリモートマシンに接続するために、Silk Test Workbench が Information Service や Open Agent との通信に使用するポートを設定できます。

Open Agent が起動すると、Silk Test Workbench およびテストするアプリケーションに対して、使用可能なポートがランダムに割り当てられます。ポート番号は *Silk Test Information Service* に登録されます。

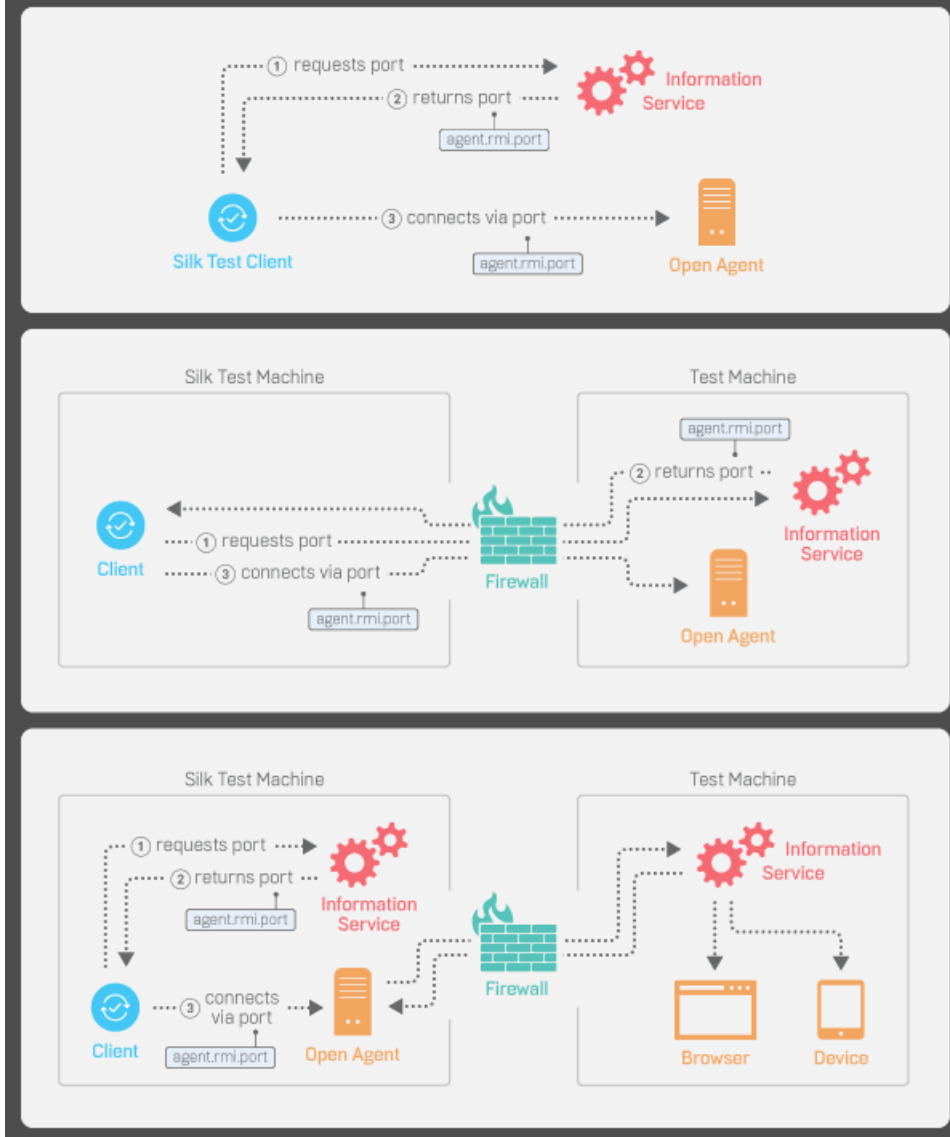
Information Service は次の情報を Silk Test Workbench に提供します。

- Silk Test Workbench が Open Agent との接続に使用できるポート番号。通信は、Silk Test Workbench とエージェントとの間で直接行われます。リモートエージェントを使用するシナリオでは、たとえばファイアウォールによって遮断されないためなど、このポートを設定する必要がある場合があります。
- Information Service がインストールされているマシン上で利用可能なブラウザ。
- Information Service がインストールされているマシンに接続しているモバイルデバイス。
- Information Service がインストールされているマシン上で利用可能なエミュレータ。
- 上記のモバイルデバイスやエミュレータ上で利用可能なモバイルブラウザ。

デフォルトでは、HTTPS ポート 48561 を使用して Open Agent は Information Service と通信します。デフォルトポートが利用可能でない場合に機能する代替ポートとして、Information Service の追加のポートを構成できます。デフォルトでは、Information Service は、代替ポートとして 2966、11998、および 11999 を使用します。

大抵の場合、手動でポート番号を設定する必要はありません。ただし、リモートマシン上でテストを実行する場合、ポート番号が競合したり、Silk Test Workbench をインストールしたマシンとテストマシン間でファイアウォールによって遮断される場合など、Silk Test Workbench とリモートマシン上の Information Service 間の通信に使用するポート番号を設定できます。テストするリモートマシンが複数ある場合は、リモートマシンごとに異なるポート番号を使用することも、すべてのリモートマシンで同じポート番号を使用することもできます。

Silk Test Client/Open Agent Connection



Open Agent に接続するポートの構成

このタスクを開始する前に、Silk Test Open Agent を停止します。

大抵の場合、手動でポート番号を設定する必要はありません。Information Service はポート構成を自動的に処理します。Open Agent との接続には、Information Service のデフォルトのポートを使用します。これにより、Information Service によって、Open Agent が使用するポートに通信が転送されます。ただし、ファイアウォールによって遮断されないようにする場合は、特定のポートに固定して通信するように Information Service ポートを設定できます。


必要に応じて、Silk Test クライアントまたはテストするアプリケーションが Open Agent に接続するために使用するポート番号を変更します。

1. agent.properties.sample ファイルに移動し、開きます。

デフォルトでは、このファイルは次の場所にあります：`%APPDATA%\Silk\SilkTest\conf`。大抵の場合、`C:\Users<ユーザー名>\AppData\Silk\SilkTest\conf` になります。ここで、<ユーザー名> は、現在のユーザー名に一致します。

2. 代替ポートの値を変更します。


大抵の場合、ポートの競合を解決するためにポートの設定を構成します。

 **注:** 各ポート番号は一意でなければなりません。エージェントのポート番号が Information Service のポート設定とは異なることを確認してください。

ポート番号は、1 から 65535 の間の任意の数値を指定できます。

ポートの設定には次のものがあります：

- agent.vtadapter.port：テストの実行時に、Silk Test Workbench と Open Agent 間の通信を制御します。
- agent.xpmodule.port：テストの実行時に、Silk Test Classic とエージェント間の通信を制御します。
- agent.autcommunication.port：Open Agent とテストするアプリケーション間の通信を制御します。
- agent.rmi.port：Open Agent と Silk4J 間の通信を制御します。
- agent.ntfadapter.port：Open Agent と Silk4NET 間の通信を制御します。
- agent.heartbeat.port：リモートマシンにインストールした Open Agent を使ってテストする場合に必要です。

 **注:** Apache Flex のテスト時に使用されるポートは、この構成ファイルでは制御できません。Flex アプリケーションのテストで割り当てられるポート番号は、6000 から始まり、各 Flex アプリケーションがテストされる度に 1 ずつ増加していきます。Flex テスト用に開始ポートを構成することはできません。

3. ファイルを agent.properties という名前で保存します。

4. Open Agent、Silk Test クライアント、およびテストするアプリケーションを再起動します。

Information Service に接続するポートの構成

このタスクを開始する前に、Silk Test Open Agent を停止します。

大抵の場合、手動でポート番号を設定する必要はありません。Information Service はポート構成を自動的に処理します。Open Agent との接続には、Information Service のデフォルトのポートを使用します。これにより、Information Service によって、Open Agent が使用するポートに通信が転送されます。ただし、ファイアウォールによって遮断されないようにする場合は、特定のポートに固定して通信するように Information Service ポートを設定できます。

デフォルトでは、Silk Test Workbench と Information Service との接続に使用されるポートは、セキュアな HTTPS 接続のポートで 48561 です。デフォルトのポートが使用可能であれば、ポート番号を指定せずに単純に hostname だけを入力できます。ポート番号を指定する場合には、Information Service のデフォルトのポートまたは追加したポートの 1 つと一致していることを確認してください。間違ったポートが指定されていると、通信に失敗します。

必要に応じて、Information Service に接続するためにすべてのクライアントが使用するポート番号を変更できます。

1. infoservice.properties.sample ファイルに移動し、開きます。

- Microsoft Windows システムでは、このファイルは C:¥ProgramData¥Silk¥Silk Test¥conf にあります。ここで、「C:¥ProgramData」は、Windows システムでデフォルトで設定されている環境変数 ALLUSERSPROFILE の値です。
- macOS では、このファイルは /Users/<ユーザー>/.silk/silktest/conf にあります。

このファイルには、コメントとサンプルの代替ポート設定が含まれています。

2. Silk Test Workbench と Information Service との通信に、HTTPS によるセキュアな接続を使用するかどうかを指定します。

- HTTPS によるセキュアな通信を使用する場合は、infoservice.https.enabled を true に設定します。これは、デフォルトの設定です。

- HTTPS によるセキュアな通信を使用しない場合は、`infoservice.https.enabled` を `false` に設定します。
3. 省略可能：HTTPS によるセキュアな通信を使用するように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、`infoservice.default.https.port` を設定します。
デフォルトの HTTPS ポートは 48561 です。ポート番号は、1 から 65535 の間の任意の数値を指定できます。
 4. 省略可能：HTTPS によるセキュアな通信を使用しないように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、`infoservice.default.port` を設定します。
デフォルトのポートは 22901 です。
 5. ファイルを `infoservice.properties` という名前で保存します。
 6. Open Agent、Silk Test クライアント、およびテストするアプリケーションを再起動します。

Silk Test Information Service プロパティの編集

`infoservice.properties` ファイルを使用して、Silk Test Information Service のポートや、HTTPS によるセキュアな接続を使用するかどうかを指定したり、Silk Test Information Service を実行しているマシン上で Silk Test がテストを実行するたびに毎回適用されるケイパビリティ (Capabilities) を指定できます。

1. `infoservice.properties.sample` ファイルがあるディレクトリに移動します。
 - Windows マシン上で、`%PROGRAMDATA%\Silk\SilkTest\conf` に移動します (C:\ProgramData\Silk\SilkTest\conf など)。
 - macOS 上で、`~/silk/silktest/conf/` に移動します。
2. `infoservice.properties.sample` ファイルの名前を `infoservice.properties` に変更します。
3. Silk Test Workbench と Information Service との通信に、HTTPS によるセキュアな接続を使用するかどうかを指定します。
 - HTTPS によるセキュアな通信を使用する場合は、`infoservice.https.enabled` を `true` に設定します。これは、デフォルトの設定です。
 - HTTPS によるセキュアな通信を使用しない場合は、`infoservice.https.enabled` を `false` に設定します。
4. 省略可能：HTTPS によるセキュアな通信を使用するように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、`infoservice.default.https.port` を設定します。
デフォルトの HTTPS ポートは 48561 です。ポート番号は、1 から 65535 の間の任意の数値を指定できます。
5. 省略可能：HTTPS によるセキュアな通信を使用するように指定した場合に、すべての HTTP リクエストを HTTPS ポートにリダイレクトするには、`infoservice.http-to-https.enabled` を `true` に設定します。
デフォルト値は、`false` です。
6. 省略可能：HTTPS によるセキュアな通信を使用しないように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、`infoservice.default.port` を設定します。
デフォルトのポートは 22901 です。
7. 省略可能：Silk Test が HTTPS 接続で使用する証明書をユーザーの証明書で置き換える場合は、[「Information Service との HTTPS 接続に使用する証明書の置換」](#)を参照してください。
8. ケイパビリティを指定するには、次の行を `infoservice.properties` ファイルに追加します。
`customCapabilities=<custom_capability_1>;<custom_capability_2>;...`

例：指定した言語での iOS シミュレータの実行

Mac 上で iOS シミュレータを常に同じ言語で実行するには、カスタム ケイパビリティ *language* および *locale* を指定します。たとえば日本語の場合、次の行を `infoservice.properties` ファイルに追加します。

```
customCapabilities=language=ja;locale=ja_JP
```

Information Service との HTTPS 接続に使用する証明書の置換

Silk Test Workbench と Information Service 間でセキュアな HTTPS 接続を使用する場合、次の自己署名証明書ファイルが使用されます。

- keystore 証明書ファイルは、Information Service HTTPS サーバーで使用されます。
- 次の証明書ファイルは、Silk Test Workbench クライアントを実行するマシンで使用されます。
 - cacerts
 - cacerts.p12
 - cacerts.pem

OpenSSL と Java keytool 実行可能ファイルを使って、これらのファイルをユーザー独自の証明書ファイルで置き換えることができます。

1. 使用するマシン上に OpenSSL と JDK がインストールされていることを確認してください。
2. JDK インストール フォルダの bin フォルダにある Java keytool 実行可能ファイルを実行します。
3. Information Service HTTPS サーバー上で、秘密キーと公開キーのペアを秘密 keystore ファイルに作成します。

```
keytool -genkey -alias jetty -keyalg RSA -keypass Borland -storepass Borland -keystore keystore -validity 1095
```

4. プロンプトの指示に従い、氏名を入力し、ホストにはワイルドカード * を入力します。
5. 秘密 keystore ファイルから、server.cer という名前のテンポラリ証明書ファイルに情報をエクスポートします。

```
keytool -export -alias jetty -storepass Borland -file server.cer -keystore keystore
```

このテンポラリ証明書ファイルは、Silk Test Workbench クライアントを実行するマシン用の証明書ファイルを生成するために必要になります。

6. server.cer ファイルから、cacerts という名前の証明書ファイルを作成します。

```
keytool -import -v -trustcacerts -alias jetty -file server.cer -keystore cacerts -keypass Borland -storepass Borland
```

7. cacerts ファイルから、cacerts.p12 という名前のテンポラリ証明書ファイルに情報をインポートします。

```
keytool -importkeystore -srckeystore cacerts -destkeystore cacerts.p12 -srcstoretype JKS -deststoretype PKCS12 -srcstorepass Borland -deststorepass Borland
```

8. テンポラリ証明書ファイル cacerts.p12 から、PKCS12 キーストア形式の公開キーストア ファイル cacerts.pem を作成します。


```
openssl pkcs12 -in cacerts.p12 -out cacerts.pem -cacerts -nokeys
```

9. ファイル (keystore、cacerts、cacerts.p12、cacerts.pem) を設定フォルダにコピーします。

- Windows マシンの場合、`%PROGRAMDATA%\SilkTest\conf` にファイルをコピーします (C: \ProgramData\SilkTest\conf など)。
- macOS の場合、`~/ .silk/silktest/conf/` にファイルをコピーします。

Open Agent を使用したリモート テスト

リモート マシン上に Silk Test Workbench をインストールして、ローカル マシンにインストールされている Silk Test Workbench からリモート ロケーション上のアプリケーションをテストすることができます。

 **注:** リモート マシンに接続されているモバイル デバイスや、リモート マシン上のエミュレータまたはシミュレータ上のモバイル アプリケーションをテストする場合や、Apple Safari またはリモート Microsoft Edge 上の Web アプリケーションをテストする場合には、リモート Open Agent ではなく、リモート Silk Test Information Service を使用する必要があります。

リモート Open Agent を使用したテスト

Silk Test Workbench を使用してリモート マシン上のアプリケーションに対してテストを再生するには、次のタスクを実行します。

1. ローカル マシン上のアプリケーションに対してテストを作成します。
2. リモート マシン上に Open Agent をインストールします。
詳細については、『[Silk Test インストール ガイド](#)』を参照してください。
3. リモート マシン上で Open Agent を開始します。

オブジェクト マップ

オブジェクト マップはテスト資産の一種であり、コントロールまたはウィンドウのロケーターではなく、コントロールまたはウィンドウに論理名 (エイリアス) を関連付ける項目が含まれています。コントロールがオブジェクト マップ資産に登録されると、ビジュアルテストおよびスクリプトでのそのコントロールに対する参照はすべて、実際のロケーター名ではなく、そのエイリアスによって行われます。

複数のスクリプトで頻繁に使用するオブジェクトを格納するために、オブジェクト マップを使用できます。複数のテストで 1 つのオブジェクト マップ項目の定義を参照できるため、ユーザーがそのオブジェクト マップ定義を 1 回更新すると、オブジェクト マップ定義を参照するすべてのテストでそのオブジェクト マップ定義が Silk Test Workbench によって更新されます。

スクリプトで、オブジェクト マップ識別子とロケーターを混在させることができます。この機能により、オブジェクト マップを比較的小さいまま保ち、管理しやすくすることが可能です。共通で使用されるオブジェクトをオブジェクト マップに格納し、まれにしかしよしないオブジェクトを参照するにはロケーターを使用します。



ヒント: オブジェクト マップが提供する機能を最適に使用するには、テストしたいアプリケーションごとに個々のプロジェクトを Silk Test Workbench に作成します。

オブジェクト マップの例

以下の構成では、ロケーターが使用されている `BrowserWindow` の定義が示されています。

```
_desktop.BrowserApplication("cnn_com").BrowserWindow("//  
BrowserWindow[1]")
```

オブジェクト マップ資産の名前は `cnn_com` です。オブジェクト マップのエイリアスによって置き換えることができるロケーターは、以下のとおりです。

```
//BrowserWindow[1]"
```

`BrowserWindow` のオブジェクト マップ エントリは `BrowserWindow` です。

結果的に、スクリプト内の `BrowserWindow` の定義は以下のようになります。

```
_desktop.BrowserApplication("cnn_com").BrowserWindow("BrowserWindow")
```

ロケーターのインデックスが変更された場合、テスト スクリプトのロケーターのすべての外観を変更する必要はなく、オブジェクト マップのエイリアスを変更するだけで済みます。Silk Test Workbench によって、オブジェクト マップ定義を参照するすべてのテストが更新されます。

オブジェクト マップ識別子とロケーターを混在させる例

つぎのサンプルコードは、オブジェクト マップ識別子と、オブジェクト マップに格納されたオブジェクトのまれに使用される子オブジェクトを指定するロケーターを混在させる方法を示します：

```
// VB  
Window window = _desktop.Window("MyApplication") // object map id -  
the application window is used often  
MenuItem aboutMenuItem = _desktop.MenuItem("@caption='About'") //  
locator - the About dialog is only used once  
aboutMenuItem.Select()
```


オブジェクト マップを使用する利点

オブジェクト マップには、以下の利点があります。


- オブジェクト マップ項目のロケーターに加えられた変更を、対応するオブジェクト マップ項目を含むすべてのテストに適用することによって、テストのメンテナンスが簡単になる。
- 大規模な機能テスト環境において、ロケーターの扱いが容易になる。
- 個々のスクリプトから独立して管理することができるようになる。
- 複雑なロケーター名がわかりやすい名前でも置き換えられるため、ビジュアル テストとスクリプトが読みやすくなる。
- テスト アプリケーションが変更された場合に変わる可能性のあるロケーターに依存しなくなる。


オブジェクト マップのオン/オフの切り替え

記録時に Silk Test Workbench でロケーター名またはオブジェクト マップのエイリアスのいずれを使用するかを設定できます。

記録中にオブジェクト マップからエイリアスを使用するには、以下を実行します。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの 記録 の隣にあるプラス記号 (+) をクリックします。記録 オプションが右側のパネルに表示されます。
3. ロケーター をクリックします。
4. オブジェクト マップ エントリを記録するか、XPath ロケーターを記録するかを定義するには、**オブジェクト マップ エントリを記録する** リストから適切な記録モードを選択します。
 - **オブジェクト マップ エントリ (新しいオブジェクトと既存のオブジェクト)**。これはデフォルトのモードです。
 - **XPath ロケーター (新しいオブジェクトと既存のオブジェクト)**。
 - **XPath ロケーター (新しいオブジェクトのみ)**。オブジェクト マップに既に存在するオブジェクトに対しては、オブジェクト マップ エントリが再利用されます。この設定を選択すると、AUT のメイン コントロールに対するオブジェクト マップを作成し、AUT に対して追加のテストを作成する間、これらのオブジェクト マップを保持することができます。

 **注:** XPath 属性のほかに、ロケーターの記録中にオブジェクト マップをマージする際に、Silk Test Workbench は要素の追加の属性を使用します。ただし、記録したスクリプトでオブジェクト マップ ID の用法を曖昧にする可能性のある属性は既存のオブジェクト マップ エントリにロケーターをマップするために使用されません。

 **注:** **オブジェクト マップを記録する** 設定を有効にすると、Silk Test Workbench 全体にわたって、ロケーター名の代わりにオブジェクト マップの項目名が表示されます。たとえば、**プロパティ** ペインで **アプリケーション構成** カテゴリを表示する場合、ロケーター名ではなくオブジェクト マップ項目名が **ロケーター** ボックスに表示されます。

オブジェクト マップのスコープ

追加のビジュアル テストまたはスクリプトを記録する場合、Silk Test Workbench は現在のプロジェクトを検索して、記録時に使用する一致したオブジェクト マップ項目が決まります。一致する項目が見つからない場合、Silk Test Workbench はすべての参照プロジェクトを検索します。一致する項目が参照プロジェクトで見つかり、その項目が使用されます。一致する項目が参照プロジェクトで見つからない場合、Silk Test Workbench によって、現在のプロジェクトに新しい項目が作成されます。



注: 一致するアイテムが参照プロジェクトに存在する場合、このルールの例外処理が実行されます。新しいオブジェクトが識別され、そのオブジェクトが参照プロジェクトのオブジェクト マップの子オブジェクトである場合は、参照プロジェクトのオブジェクト マップに追加されます。

オブジェクト マップで現在のプロジェクトのみを使用するには、**オプション** ダイアログ ボックスの **オブジェクト マップの場所** オプションを **現在のプロジェクト** に設定します。このように設定すると、一致するオブジェクト マップ項目の検索および新しい項目の記録で、現在のプロジェクトのみが使用されます。

操作の記録中でのオブジェクト マップのマージ

Silk Test Workbench を使用して操作を記録するときに、Silk Test Workbench は、既存のオブジェクト マップ エントリが再利用できるかどうか確認します。Silk Test Workbench は、新しいロケーターが生成されるときに、記録中に直接確認します。Silk Test Workbench は、テスト対象アプリケーションで現在記録されているオブジェクトが既存のオブジェクト マップ エントリと完全に一致するかどうか確認し、一致する場合に Silk Test Workbench はオブジェクト マップからそのオブジェクト マップ識別子を再利用します。

この動作には以下のような利点があります。

- オブジェクト マップのロケーターが変更された場合でも、Silk Test Workbench は記録中にオブジェクト マップ識別子を正しく再利用します。
- 記録したスクリプトに間違ったオブジェクト マップ識別子を含むはずがないため、間違ったオブジェクト マップ識別子によって再生に失敗することは決してありません。
- 階層のレベルをさらに追加した場合など、オブジェクト マップを再構成した場合でも、オブジェクト マップ識別子を再利用することができます。

例

Micro Focus Web サイト (<http://www.borland.com>) の **Products** リンクをクリックしたとき、Silk Test Workbench は、次のスクリプトを記録します。

```
With _desktop.BrowserApplication( "borland_com" )
  With .BrowserWindow( "BrowserWindow" )
    .DomLink( "Products" ).Click( MouseButton .Left, New Point (47, 18))
  End With
End With
```

記録したオブジェクト マップは次のようになります。

```
borland_com //BrowserApplication
  BrowserWindow //BrowserWindow
  Products //
A[@textContents='Products']
```

ここで、Micro Focus Web サイトのヘッダー部分を含むようにオブジェクト マップを手動で再構成した場合を考えます。

```
borland_com //BrowserApplication
  BrowserWindow //BrowserWindow
  header //
HEADER[@role='banner']
  Products //
A[@textContents='Products']
```

Products リンクをクリックを記録すると、オブジェクト マップが正しく再利用され、次のスクリプトが記録されます。

```
With _desktop.BrowserApplication( "borland_com" )
  With .BrowserWindow( "BrowserWindow" )
    .DomElement("header").DomLink( "Products" ).Click( MouseButton .Left
, New Point (47, 18))
```

End With
End With



注: About リンクなどの Micro Focus Web サイトのヘッダー部分にあるほかのオブジェクトを記録すると、Silk Test Workbench は **header** ではなく、**BrowserWindow** の子として **About** オブジェクト マップ エントリを追加します。

Web アプリケーションでのオブジェクト マップの使用

デフォルトで、Web アプリケーションに対する操作を記録すると、Silk Test Workbench は、*Common* プロジェクトで、ネイティブ ブラウザのコントロール用に *WebBrowser* という名前のオブジェクト マップを作成し、各 Web ドメイン用にオブジェクト マップ資産を作成します。

記録 オプションの **全般** セクションで、**オブジェクト マップの場所** を現在のプロジェクトに変更できます。その後、Silk Test Workbench は記録中に Web ドメインごとにオブジェクト マップ資産を生成します。

印刷または設定用のメイン ウィンドウやダイアログ ボックスなど、Web ドメインに特有ではない共通のブラウザ コントロールの場合、*WebBrowser* という名前を使用して、現在のプロジェクトに追加のオブジェクト マップが生成されます。生成されたオブジェクト マップ資産は、**アセット ブラウザ**で確認できません。

オブジェクト マップで、オブジェクト マップのエントリのグループ化に使用される URL パターンを編集できます。パターンを編集すると、Silk Test Workbench はそのパターンの構文検証を行います。パターンには、ワイルドカード * および ? を使用できます。

例

<http://www.borland.com> および <http://www.microfocus.co.jp> で何らかの操作を記録した後、プリンタ ダイアログを開くと、次の 3 つの新しいオブジェクト マップ資産が**アセット ブラウザ**に追加されます。

- WebBrowser
- borland_com
- microfocus_com



注: Silk Test Workbench では、オブジェクト マップのないプロジェクトに対してのみ、新しいオブジェクト マップ資産が生成されます。バージョン 14.0 よりも前のバージョンの Silk Test Workbench を使用して生成されたオブジェクト マップをすでに含む Silk Test Workbench の Web アプリケーションに対する操作を記録すると、追加で記録されたエントリは既存のオブジェクト マップに保存されます。Web ドメインに対して追加のオブジェクト マップ資産が生成されることはありません。

オブジェクト マップ項目名の変更

オブジェクト マップでは、項目とロケーターの名前を手動で変更できます。



警告: オブジェクト マップ項目の名前を変更すると、その項目を使用するすべてのビジュアル テストまたはスクリプトが影響を受けます。たとえば、**キャンセル** ボタンのオブジェクト マップ項目の名前を **CancelMe** から **Cancel** に変更すると、**CancelMe** を使用するすべてのスクリプトを、**Cancel** を使用するよう手動で変更する必要があります。

オブジェクト マップ項目は一意である必要があります。重複するオブジェクト マップ項目を追加しようとすると、オブジェクト マップ項目は一意である必要があることが Silk Test Workbench から通知されます。

無効な文字またはロケータを使用すると、項目名またはロケータ テキストが赤で表示され、ツール ヒントにエラーの説明が表示されます。オブジェクト マップ項目として無効な文字は、¥、/、<、>、"、:、*、?、|、=、..、@、[,]です。無効なロケータ パスは、空または不完全なロケータ パスです。

オブジェクト マップのルート ノードの名前を変更すると、オブジェクト マップ資産の名前も変更されます。ルート ノードの名前を既存のオブジェクト マップ資産の名前に変更した場合、オブジェクト マップ資産を保存することはできません。オブジェクト マップ資産の名前を変更すると、オブジェクト マップのルート ノードの名前も変更されます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- 名前を変更するオブジェクト マップ項目を含むオブジェクト マップをダブルクリックします。
- 名前を変更するオブジェクト マップ項目を含むオブジェクト マップを右クリックし、**開く** を選択します。

オブジェクト マップ項目および各項目に関連付けられたロケータの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. 名前を変更するオブジェクト マップ項目に移動します。

たとえば、名前を変更する項目を検索するには、ノードの展開が必要な場合があります。

6. 名前を変更するオブジェクトをクリックしてから、オブジェクトを再度クリックするか、**F2** を押します。

インライン編集モードにアクセスすると、オブジェクトを 1 回クリックして編集できます。**Tab** を押して、オブジェクト マップ項目の名前と説明を切り替えます。

項目フィールドの周囲に黒のボックスが表示され、テキストが強調表示されます。

7. 使用する項目名を入力し、Enter を押します。

無効な文字を使用すると、項目名が赤で表示されます。

新しい名前が **項目名** リストに表示されます。

8. 次のいずれか 1 つのステップを行います：

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。


- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。



注: オブジェクト マップ ツリーに含まれるすべてのノードのすべての子ノードは、オブジェクト マップを保存するときにアルファベット順にソートされます。

変更した項目名を既存のビジュアルテストまたはスクリプトで使用する場合は、新しい項目名を使用するようにビジュアルテストまたはスクリプトを手動で変更する必要があります。

 **注:** Web アプリケーションまたはモバイル Web アプリの記録中に、**操作の選択** ダイアログでオブジェクト マップ エントリの名前を直接変更できます。オブジェクトを右クリックして、**操作の選択** ダイアログの **オブジェクトの識別** 領域を展開します。そして、**オブジェクト マップ ID** フィールドでオブジェクト マップ エントリを編集できます。この機能は、次のブラウザーに対するテストで利用できます。

- Microsoft Edge
- Apple Safari
- Mozilla Firefox 41 以降
- Google Chrome 50 以降
- モバイル ブラウザー

オブジェクト マップの変更

デフォルトでは、オブジェクト マップは読み取り専用です。オブジェクト マップを使用するビジュアルテストまたはスクリプトを記録すると、オブジェクト マップ項目が資産としてデータベースに追加されます。

オブジェクト マップを使用するビジュアルテストまたはスクリプトの記録が完了すると、別のユーザーがそのオブジェクト マップを使用しているかどうか Silk Test Workbench で判断されます。別のユーザーがオブジェクト マップを変更していると、その旨を通知するメッセージ ボックスが表示され、変更しているユーザーの名前が示されます。ロックされたオブジェクト マップによって記録できなくなるのを防ぐには、**オプション** ダイアログ ボックスの **ロック済みのオブジェクト マップに対してロケータを記録する オプション** を **はい** に設定します。このように設定すると、別のユーザーがオブジェクト マップを変更している場合に、以前に識別されたオブジェクト マップ項目が適用可能な場合に使用され、新しいオブジェクトを記録する場合にはロケータが使用されます。

既存のオブジェクト マップは、オブジェクト マップに構造化要素をさらに追加したとしても、記録中に既存のオブジェクト マップ識別子を再利用することができます。

例：既存のオブジェクト マップへの DIV の追加

次の単純なオブジェクト マップの email フィールドと login フィールドをまとめる DIV 要素を追加することを考えます。

```
demo_borland_com //BrowserApplication
  BrowserWindow //
BrowserWindow
  login-form email //
INPUT[@id='login-form:email']
  login-form login //
INPUT[@id='login-form:login']
```

新たに DIV *loginArea* を追加することにより、オブジェクト マップの構造を変更できますが、オブジェクト マップは、記録中にオブジェクト マップ識別子は正しく再利用することができます。

```
demo_borland_com //BrowserApplication
  BrowserWindow //
BrowserWindow
loginArea //DIV[@id='loginArea']
  login-form email //
INPUT[@id='login-form:email']
```

```
login-form login //
INPUT[@id='login-form:login']
```

オブジェクト マップのロケーターの変更

ビジュアルテストまたはスクリプトを記録するときに、ロケーターは自動的にオブジェクト マップ項目に関連付けられます。ただし、より汎用的にするために、ロケーター パスを変更できます。たとえば、テストアプリケーションで特定のコントロールに自動的に日付または時刻が割り当てられる場合、ワイルドカードを使用するようにそのコントロールのロケーターを変更できます。ワイルドカードを使用すると、それぞれのテストで異なる日付または時刻が挿入される場合でも、各テストで同じロケーターを使用できます。

1. **表示** > **アセットブラウザ** を選択するか、**開始画面** の **入門ガイド** > **アセットブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- 変更するロケーターを含むオブジェクト マップをダブルクリックします。
- 変更するロケーターを含むオブジェクト マップを右クリックし、**開く** を選択します。

オブジェクト マップ項目および各項目に関連付けられたロケーターの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. 変更するロケーターに移動します。
たとえば、変更するロケーターを検索するには、ノードの展開が必要な場合があります。
6. 変更するロケーター パスをクリックしてから、ロケーター パスを再度クリックするか、**F2** を押します。

インライン編集モードにアクセスすると、ロケーター パスを 1 回クリックして編集できます。**Tab** を押して、オブジェクト マップ項目の名前と説明を切り替えます。

ロケーター パス フィールドの周囲に黒のボックスが表示され、テキストが強調表示されます。

7. 有効なロケーター パスがある場合は、使用する項目名とロケーター パスを入力して **Enter** を押すことができます。有効なロケーター パスを判別するには、以下のステップで説明するように、**オブジェクトの識別** ダイアログ ボックスを使用します。

- a) **ツール** > **オブジェクトの識別** を選択します。

テストアプリケーションがオブジェクト マップに関連付けられていないと、**アプリケーションの選択** ダイアログ ボックスが表示されることがあります。この場合は、テストするアプリケーションを選択し、**OK** をクリックします。

オブジェクトの識別 ダイアログ ボックスが開きます。

- b) **選択モード** を指定します。

- **クリック** : オブジェクトをクリックして、ロケータを識別します。
- **ホット キー** : **キー入力** リスト ボックスで指定されたキー入力の組み合わせを使用してオブジェクトをキャプチャするには、このモードを指定します。通常、このモードは、ユーザーがクリックしたときにだけ表示されるメニューやコンボ ボックスなどのオブジェクトをキャプチャする場合に選択します。このモードでは、オブジェクトを選択し、ホット キー入力の組み合わせを押すことによって、オブジェクトを閉じることなくロケータをキャプチャします。

c) **識別の開始** をクリックします。

d) 記録したいオブジェクト上にマウスを移動して、次のステップのいずれかを行います :

- **クリック** モードを使用する場合は、識別するオブジェクトをクリックします。
- キー入力の組み合わせを押して、**ホット キー** モードでオブジェクトをキャプチャします。
デフォルトでは、キー入力の組み合わせは Ctrl+Shift です。

Silk Test Workbench によって、関連するロケータ文字列が **選択済みロケータ** テキスト ボックスに表示されます。

e) **ロケータの詳細** テーブルで、使用するロケータを選択します。新しいロケータは、**選択済みロケータ** テキスト ボックスに表示されます。

f) **貼り付け** をクリックします。Silk Test Workbench によって、項目名および関連付けられたロケータがオブジェクト マップに追加されます。

8. 必要に応じて、ニーズに合わせて項目名またはロケータ テキストを変更します。

無効な文字またはロケータを使用すると、項目名またはロケータ テキストが赤で表示され、ツール ヒントにエラーの説明が表示されます。

オブジェクト マップ項目として無効な文字は、¥、/、<、>、"、:、*、?、|、=、.、@、[,] です。

無効なロケータ パスは、空または不完全なロケータ パスです。

9. 次のいずれか 1 つのステップを行います :

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。

- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。

変更したロケータ パスが既存のビジュアル テストまたはスクリプトによって使用されている場合は、新しいロケータ パスを使用するように、そのビジュアル テストまたはスクリプトを手動で変更する必要があります。

オブジェクト マップの比較

2 つの異なるオブジェクト マップや、同じオブジェクト マップの 2 つのバージョンを比較して、これらの間の相違点を調べることができます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 調査したいオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 2 つの異なるオブジェクト マップを比較する場合は、**Ctrl + クリック** を使って比較する 2 つのオブジェクト マップを選択します。
4. 同じオブジェクト マップの 2 つのバージョンを比較するには :

- a) 調査したいオブジェクト マップを右クリックして、**すべてのバージョンを表示** を選択します。
- b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する 2 つのバージョンを選択します。


5. 相違点の表示 をクリックします。

- 変更箇所は黄色で表示されます。
- 追加箇所は緑色で表示されます。
- 比較した資産の一方に存在しない項目はグレーで表示されます。

オブジェクト マップのマージ

2 つの異なるオブジェクト マップや、同じオブジェクト マップの 2 つのバージョンを比較して、オブジェクト マップ間、またはオブジェクト マップのバージョン間の相違点をマージすることができます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。


 **ヒント:** 調査したいオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 2 つの異なるオブジェクト マップを比較する場合は、**Ctrl + クリック** を使って比較する 2 つのオブジェクト マップを選択します。
4. 同じオブジェクト マップの 2 つのバージョンを比較するには：
 - a) 調査したいオブジェクト マップを右クリックして、**すべてのバージョンを表示** を選択します。
 - b) **資産のバージョン** ペインで、**Ctrl + クリック** を使って比較する 2 つのバージョンを選択します。
5. 相違点の表示を移動するには、ツールバーの **次の相違点** と **前の相違点** を使用します。
6. 選択した相違点をマージするには：

- **左側にコピー** をクリックすると、右側のペインに表示されているバージョンまたは資産の選択した相違点を、左側に表示されているバージョンまたは資産にマージします。
- **右側にコピー** をクリックすると、左側のペインに表示されているバージョンまたは資産の選択した相違点を、右側に表示されているバージョンまたは資産にマージします。

ID とロケーターの両方が異なるオブジェクト マップのエントリをマージする場合は、常に両方とも他方の資産にコピーされます。ID またはロケーターだけをマージする方法はありません。ただし、ルートオブジェクト マップのエントリをマージする場合は、ロケーターだけが他方の資産にコピーされます。

- 変更箇所は黄色で表示されます。
- 追加箇所は緑色で表示されます。
- 比較した資産の一方に存在しない項目はグレーで表示されます。

 **注:** Silk Test Workbench ユーザーが資産のアクティブ バージョンを比較またはマージしている間、その資産はロックされ、他のユーザーは編集できなくなります。資産が他のウィンドウで編集用に既に開かれている場合に、その資産のアクティブ バージョンを比較またはマージしようとする、読み取り専用モードでそのバージョンが開かれます。読み取り専用かどうかは、そのバージョンのタイトルバーに表示されます。

テスト アプリケーションからのオブジェクト マップの更新

テスト アプリケーションの項目が変化した場合は、**オブジェクト マップ** UI を使用してそれらの項目のロケーターを更新できます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- 使用するオブジェクト マップをダブルクリックします。
- 使用するオブジェクト マップを右クリックし、**開く** をクリックします。

オブジェクト マップ項目および各項目に関連付けられたロケーターの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. **ロケータの更新** をクリックします。**オブジェクトの識別** ダイアログ ボックス が表示され、Silk Test Workbench によってテスト アプリケーションが開かれます。

6. 記録するオブジェクトの上にカーソルを合わせて、**Ctrl+Alt** を押します。Silk Test Workbench の **ロケーター** テキスト フィールドにロケーター文字列が表示されます。

7. **ロケーターの詳細** テーブルで、使用するロケーターを選択します。

8. **ロケーター** テキスト フィールドに表示されているロケーターから、使用しない属性を削除します。

9. **ロケーターの検証** をクリックして、ロケーターが機能することを検証します。

10 **ロケーターをエディターに貼り付け** をクリックして、オブジェクト マップのロケーターを更新します。

11変更されたオブジェクト マップを保存します。

AUT からオブジェクト マップ項目を更新するときに、オブジェクト マップ ツリーのリーフ ノードの XPath 表現のみを変更できます。親ノードの XPath 表現を変更することはできません。オブジェクト マップ ツリー内のより高いレベルのノードにある XPath 表現が更新後に整合しなくなると、エラー メッセージが表示されます。

例

たとえば、次の 3 つの階層レベルを持つオブジェクト マップ ID を含むオブジェクト マップ項目があるとします：

WebBrowser.Dialog.Cancel

これらの階層レベルに対応する XPath 表現は次のようになります：

`/BrowserApplication//Dialog//PushButton[@caption='Cancel']`

- 最初の階層レベル： `/BrowserApplication`
- 2 番目の階層レベル： `//Dialog`
- 3 番目の階層レベル： `//PushButton[@caption='Cancel']`

次のロケーターを使用して、オブジェクト マップ項目を更新できます：

`/BrowserApplication//Dialog//PushButton[@id='123']`

- 最初の階層レベル： `/BrowserApplication`
- 2 番目の階層レベル： `//Dialog`
- 3 番目の階層レベル： `//PushButton[@id='123']`

2 番目のレベルの階層が一致しないため、次のロケータを使用してオブジェクト マップ項目を更新することはできません：

```
/BrowserApplication//BrowserWindow//PushButton[@id='9999999']
```

- 最初の階層レベル： /BrowserApplication
- 2 番目の階層レベル： //BrowserWindow
- 3 番目の階層レベル： //PushButton[@id='9999999']

オブジェクト マップ項目のコピー

オブジェクト マップ内、またはオブジェクト マップ間で、オブジェクト マップ エントリをコピーおよび貼り付けできます。たとえば、2 つの異なるテスト アプリケーションに同じ機能が存在する場合は、一方のオブジェクト マップの一部分をコピーして、他方のオブジェクト マップに貼り付けることができます。

1. **表示 > アセットブラウザ** を選択するか、**開始画面 の 入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- コピーするオブジェクト マップ項目を含むオブジェクト マップをダブルクリックします。
- コピーするオブジェクト マップ項目を含むオブジェクト マップを右クリックし、**開く** を選択します。

オブジェクト マップ項目および各項目に関連付けられたロケータの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. コピーするオブジェクト マップ項目に移動します。

たとえば、コピーするオブジェクト マップ項目を検索するには、ノードの展開が必要な場合があります。

6. 次のいずれか 1 つを選んでください：

- コピーするオブジェクト マップ項目を右クリックし、**ツリーのコピー** を選択します。
- コピーするオブジェクト マップ項目をクリックし、**編集 > コピー** を選択します。
- コピーするオブジェクト マップ項目をクリックし、Ctrl+C を押します。

7. オブジェクト マップ階層で、コピーした項目を貼り付ける位置に移動します。

たとえば、階層の第 1 レベルに項目を組み込むには、項目リストの最初の項目の名前をクリックします。特定の項目の 1 レベル下にコピーする項目の位置を設定するには、コピーする項目の上にある項目をクリックします。

オブジェクト マップ間でコピーして貼り付けるには、オブジェクト マップ項目をコピーしたマップを終了し、オブジェクト マップ項目を貼り付けるオブジェクト マップを開いて編集する必要があります。

8. 次のいずれか 1 つを選んでください：

- コピーしたオブジェクト マップ項目を貼り付けるオブジェクト マップ内の位置を右クリックし、**貼り付け** を選択します。
- コピーしたオブジェクト マップ項目を貼り付けるオブジェクト マップ内の位置をクリックし、**編集 > 貼り付け** を選択します。
- コピーしたオブジェクト マップ項目を貼り付けるオブジェクト マップ内の位置をクリックし、Ctrl +V を押します。

オブジェクト マップ間でコピーして貼り付けるには、オブジェクト マップ項目を貼り付ける前に、**編集** を必ずクリックします。

オブジェクト マップ項目が、階層内の新しい位置に表示されます。

9. 次のいずれか 1 つのステップを行います：

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。

- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。

移動したオブジェクト マップ項目を既存のビジュアル テストまたはスクリプトで使用する場合は、階層内の新しい位置を使用するようにビジュアル テストまたはスクリプトを手動で変更する必要があります。

オブジェクト マップ項目の追加

ビジュアル テストまたはスクリプトを記録すると、オブジェクト マップ項目が自動的に作成されます。場合によっては、手動でオブジェクト マップ項目を追加することもできます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 新しい項目を追加したい場所で、オブジェクト マップをダブルクリックします。オブジェクト マップ項目および各項目に関連付けられたロケーターの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. オブジェクト マップ階層で、新しいオブジェクト マップ項目を追加したい位置の下の項目を右クリックします。

たとえば、階層の第 1 レベルに項目を組み込むには、項目リストの最初の項目の名前を右クリックします。特定の項目の 1 レベル下に新しい項目の位置を設定するには、新しい項目を配置したい位置の下の項目をクリックします。

6. **新規挿入** をクリックします。新しい項目が、現在のノードの最初の子として階層に追加されます。

7. 有効なロケータ パスがある場合は、使用する項目名とロケータ パスを入力して Enter を押すことができます。有効なロケータ パスを判別するには、以下のステップで説明するように、**オブジェクトの識別** ダイアログ ボックスを使用します。

a) ツール > **オブジェクトの識別** を選択します。

テスト アプリケーションがオブジェクト マップに関連付けられていないと、**アプリケーションの選択** ダイアログ ボックスが表示されることがあります。この場合は、テストするアプリケーションを選択し、**OK** をクリックします。

オブジェクトの識別 ダイアログ ボックスが開きます。

b) **選択モード** を指定します。

- **クリック** : オブジェクトをクリックして、ロケータを識別します。
- **ホット キー** : **キー入力** リスト ボックスで指定されたキー入力の組み合わせを使用してオブジェクトをキャプチャするには、このモードを指定します。通常、このモードは、ユーザーがクリックしたときにだけ表示されるメニューやコンボ ボックスなどのオブジェクトをキャプチャする場合に選択します。このモードでは、オブジェクトを選択し、ホット キー入力の組み合わせを押すことによって、オブジェクトを閉じることなくロケータをキャプチャします。

c) **識別の開始** をクリックします。

d) 記録したいオブジェクト上にマウスを移動して、次のステップのいずれかを行います :

- **クリック** モードを使用する場合は、識別するオブジェクトをクリックします。
- キー入力の組み合わせを押して、**ホット キー** モードでオブジェクトをキャプチャします。

デフォルトでは、キー入力の組み合わせは Ctrl+Shift です。

Silk Test Workbench によって、関連するロケータ文字列が **選択済みロケータ** テキスト ボックスに表示されます。

e) **ロケータの詳細** テーブルで、使用するロケータを選択します。新しいロケータは、**選択済みロケータ** テキスト ボックスに表示されます。

f) **貼り付け** をクリックします。Silk Test Workbench によって、項目名および関連付けられたロケータがオブジェクト マップに追加されます。

8. 必要に応じて、ニーズに合わせて項目名またはロケータ テキストを変更します。

無効な文字またはロケータを使用すると、項目名またはロケータ テキストが赤で表示され、ツール ヒントにエラーの説明が表示されます。

オブジェクト マップ項目として無効な文字は、¥、/、<、>、"、:、*、?、|、=、.、@、[,] です。

無効なロケータ パスは、空または不完全なロケータ パスです。


9. 次のいずれか 1 つのステップを行います :

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。

- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。

 **注:** オブジェクト マップ ツリーに含まれるすべてのノードのすべての子ノードは、オブジェクト マップを保存するときにアルファベット順にソートされます。

スクリプトからオブジェクト マップを開く

スクリプトを編集している際に、スクリプトのオブジェクト マップ エントリを右クリックし、**資産を開く** を選択してオブジェクト マップを開くことができます。オブジェクト マップは GUI 上で開かれます。

例

```
// VB .NET code
<TestMethod()> Public Sub TestMethod1()
    With _desktop.Window("Untitled -
Notepad").TextField("TextField").TypeKeys("hello")
    End With
End Sub
```

上記のコード例で、Untitled - Notepad エントリをオブジェクト マップで開く場合は、Untitled - Notepad を右クリックします。オブジェクト マップで Untitled - Notepad.TextField エントリをオブジェクト マップで開く場合は、TextField を右クリックします。

テスト アプリケーションでのオブジェクト マップ項目の検索

オブジェクト マップ項目を追加または記録したあと、**検索** をクリックして、テスト アプリケーションで項目をハイライトできます。変更する項目であることをオブジェクト マップ内で確認する場合などに項目をハイライトできます。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- 使用するオブジェクト マップをダブルクリックします。
- 使用するオブジェクト マップを右クリックし、**開く** をクリックします。

オブジェクト マップ項目および各項目に関連付けられたロケーターの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押し、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. オブジェクト マップ階層で、テスト アプリケーションでハイライトするオブジェクト マップ項目を選択します。



注: テスト アプリケーションの 1 つのインスタンスのみが実行中であることを確認します。テスト アプリケーションの複数のインスタンスを実行すると、複数のオブジェクトがロケーターと一致するためエラーになります。

6. **検索** をクリックします。

テスト アプリケーションがオブジェクト マップに関連付けられていないと、**アプリケーションの選択** ダイアログ ボックスが表示されることがあります。この場合は、テストするアプリケーションを選択し、**OK** をクリックします。

アプリケーションの選択 ダイアログ ボックスで誤ったアプリケーションを選択した場合は、**検索** サブメニューで **アプリケーションを指定して検索** をクリックします。**アプリケーションの選択** ダイアログ ボックスが開き、正しいテスト アプリケーションを選択できます。

Silk Test Workbench によってテスト アプリケーションが開かれ、オブジェクト マップ項目を示すコントロールの周囲に緑のボックスが表示されます。

オブジェクト マップのエラーの検出

無効な文字またはロケータを使用すると、項目名またはロケータ テキストが赤で表示され、ツール ヒントにエラーの説明が表示されます。**オブジェクト マップ** ウィンドウのツール バーを使用して、エラーに移動します。

1. **表示 > アセットブラウザ** を選択するか、**開始画面 の 入門ガイド > アセットブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- トラブルシューティングするオブジェクト マップをダブルクリックします。
- トラブルシューティングするオブジェクト マップを右クリックし、**開く** を選択します。

オブジェクト マップ項目および各項目に関連付けられたロケータの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. 赤色で表示された項目名またはロケータ テキストを探します。
6. 必要に応じて、ニーズに合わせて項目名またはロケータ テキストを変更します。

無効な文字またはロケータを使用すると、項目名またはロケータ テキストが赤で表示され、ツール ヒントにエラーの説明が表示されます。

オブジェクト マップ項目として無効な文字は、¥、/、<、>、"、:、*、?、|、=、.、@、[,] です。

無効なロケータ パスは、空または不完全なロケータ パスです。

7. 次のいずれか 1 つのステップを行います：

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。

- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。

オブジェクト マップ項目の削除

テスト アプリケーションに存在しなくなったなどの理由により、オブジェクト マップから項目を削除できます。

1. **表示** > **アセットブラウザ** を選択するか、**開始画面** の **入門ガイド** > **アセットブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。



ヒント: 編集するオブジェクト マップが表示されない場合は、**アクティブプロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 削除するオブジェクト マップ項目を含むオブジェクト マップをダブルクリックします。オブジェクト マップ項目および各項目に関連付けられたロケーターの階層が、オブジェクト マップに表示されます。



注: 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押して、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. 削除するオブジェクト マップ項目に移動します。

たとえば、削除するオブジェクト マップ項目を検索するには、ノードの展開が必要な場合があります。

6. 次のいずれか 1 つを選んでください：

- 削除するオブジェクト マップ項目を右クリックし、**削除** を選択するか、そのオブジェクト マップ項目のすべての子項目も削除する場合は **ツリーの削除** を選択します。
- 削除するオブジェクト マップ項目をクリックし、**Del** を押すか、そのオブジェクト マップ項目のすべての子項目も削除する場合は **Ctrl+Del** を押します。

オブジェクト マップ項目を削除した後、フォーカスはオブジェクト マップの次の項目に移動します。

7. 次のいずれか 1 つのステップを行います：

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。

- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。


削除したオブジェクト マップ項目または子オブジェクトを既存のビジュアル テストまたはスクリプトで使用する場合は、そのオブジェクト マップ項目への参照をビジュアル テストまたはスクリプトで、手動で変更する必要があります。

オブジェクト マップを最初に書き出す

ベストプラクティスとして、テストを記録する前に、すべてのオブジェクト マップ項目を書き出し、確認することをお勧めします。


AUT のすべての利用可能な項目をもつオブジェクト マップを最初に書き出すためには、テスト対象アプリケーションのすべてのオブジェクトをクリックし、すべてのウィンドウとダイアログ ボックスを開くビジ

ユアルテストを作成する必要があります。その後、各オブジェクトに対するオブジェクト マップ項目を確認し、機能テストを記録する前に、必要な変更を加えることができます。オブジェクト マップ項目を確認し、修正した後に、オブジェクト マップを書き出すために作成したビジュアル テストを削除できます。


 **ヒント:** オブジェクト マップ中の項目間を移動するには、矢印キーを使用できます。

オブジェクト マップの要素のグループ化

オブジェクト マップの項目が一貫した親オブジェクトを持たない場合、XPath の現在の要素のロケーターを意味するロケーター "." を持つ新しいツリー項目を追加して、これらの要素をグループ化できます。

 **警告:** オブジェクト マップ項目をグループ化すると、それらの項目を使用するすべてのビジュアル テストまたはスクリプトが影響を受けます。これらの項目を使用するすべてのビジュアル テストまたはスクリプトは、新しいロケーターを使用するように手動で変更する必要があります。


1. **表示 > アセット ブラウザ** を選択するか、**開始画面 の 入門ガイド > アセット ブラウザ** を選択します。
2. **資産の種類** リストから **オブジェクト マップ** を選択します。選択したプロジェクトの既存のオブジェクト マップが、右側のペインに表示されます。

 **ヒント:** 編集するオブジェクト マップが表示されない場合は、**アクティブ プロジェクト** リストで正しいプロジェクトが選択されていることを確認してください。

3. 次のいずれか 1 つを選んでください：

- 編集するオブジェクト マップをダブルクリックします。
- 編集するオブジェクト マップを右クリックし、**開く** をクリックします。

オブジェクト マップ項目および各項目に関連付けられたロケーターの階層が、オブジェクト マップに表示されます。

 **注:** 選択したオブジェクト マップを別のユーザーが現在変更していると、そのことを通知するメッセージ ボックスが表示され、オブジェクト マップを現在ロックしているユーザーの名前が表示されます。

4. **編集** をクリックします。

デフォルトでは、オブジェクト マップは読み取り専用です。**編集** をクリックすると読み取り/書き込みアクセス権が付与され、変更中に他のユーザーがそのエントリを編集できなくなります。**F2** を押し、オブジェクト マップの現在選択中のフィールドを編集することもできます。

編集 ボタンが **元に戻す** に変わります。このセッション中に加えた変更を取り消すには、**元に戻す** をクリックします。

5. 新しく追加する構成項目のすぐ上のツリー項目を右クリックして、**新規挿入** を選択します。
6. 新しいオブジェクト マップ項目の **項目名** フィールドをダブルクリックします。
7. 使用する項目名を入力し、Enter を押します。
無効な文字を使用すると、項目名が赤で表示されます。
新しい名前が **項目名** リストに表示されます。
8. 新しいオブジェクト マップ項目の **ロケーター パス** フィールドをクリックして、フィールドに「.」を入力します。
9. Enter を押します。
- 10 新しい項目の下に新たに配置し直したいすべてのオブジェクト マップ項目に対して、次の操作を行います。
 - a) 配置し直したい項目を右クリックして、**ツリーの切り取り** を選択します。
 - b) 新しい構成項目を右クリックして、**貼り付け** を選択します。
- 11 次のいずれか 1 つのステップを行います：

- **保存** をクリックして変更を保存し、オブジェクト マップの操作を続行します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、オブジェクト マップのモードは読み取り専用に戻ります。

- **保存して閉じる** をクリックして変更を保存し、オブジェクト マップを終了します。

Silk Test Workbench により変更内容が新しいバージョンとして保存され、他のユーザーがオブジェクト マップを変更できるようになります。

オブジェクト マップ：よくある質問

このセクションでは、Silk Test Workbench でオブジェクト マップを使用する時に直面する可能性のある質問を一覧します。

複数のオブジェクト マップを単一のオブジェクト マップのマージする方法

既存のオブジェクト マップをマージせずに、同じオブジェクト マップに記録することを Micro Focus は推奨しますが、テキスト エディタを使用して複数のオブジェクト マップを単一のマップにマージすることができます。

テスト スクリプトを削除したときにオブジェクト マップで起こること

オブジェクト マップ エントリを含んだテスト スクリプトまたはビジュアル テストを削除しても、関連するオブジェクト マップは変更されません。すべてのオブジェクト マップ エントリはそのまま残ります。

テスト対象アプリケーションのオブジェクト マップを手動で作成する方法

Micro Focus では、テストの記録時にオブジェクト マップを作成することを推奨します。しかし、空のオブジェクト マップを作成し、このマップにオブジェクト マップの項目を手動で追加することもできます。

新しいオブジェクト マップを作成するには、**アセットブラウザ** の **オブジェクト マップ** を右クリックし、**オブジェクト マップの新規作成** を選択します。


イメージ解決のサポート

イメージ解決は、次の場合に使用できます。

- オブジェクト解決で識別できない、高度にカスタマイズされたコントロールを含むテスト アプリケーションを簡単に操作する場合。座標ベースのクリックの代わりにイメージ クリックを使用し、指定されたイメージをクリックできます。
- テスト対象アプリケーションのグラフィカル オブジェクト (グラフなど) をテストする場合。
- テスト対象アプリケーションの視覚的な UI のチェックを実行する場合。

認識されないコントロールをクリックするには、ImageClick メソッドとイメージ資産を使用します。テスト対象アプリケーションに、認識されないコントロールがあるかどうかを確認するには、VerifyAsset メソッドとイメージ検証を使用します。


イメージ解決メソッドは、Silk Test Workbench でサポートされるすべてのテクノロジー ドメインでサポートされます。

 **注:** イメージ解決メソッドは、画面に表示されていないコントロールに対しては機能しません。たとえば、スクロールして画面外にあるイメージに対しては、イメージ解決を使用できません。

イメージ クリックの記録

イメージ クリックの記録を行うと、大量のイメージが生成されてわかりにくくなるため、デフォルトではイメージ クリックの記録が無効となり、座標ベースのクリックの記録が優先されます。


イメージ クリックの記録を有効にするには、**ツール > オプション > 記録 > 全般** をクリックし、'**ImageClick**' を記録する の値を はい に設定します。

 **注:** モバイルブラウザを記録する場合、イメージ クリックの記録を有効にする必要はありません。

イメージ クリックの記録を有効にすると、Silk Test Workbench は、オブジェクト解決またはテキスト解決ができない場合に、ImageClick メソッドを記録します。イメージ クリックが記録されない場合でも、任意のコントロールに対してイメージ クリックをスクリプトに挿入できます。

ImageClick 操作を記録しない場合は、イメージ クリックの記録をオフに切り替えて通常のクリックまたはテキスト クリックを記録できます。

 **注:** 記録したイメージは再利用されません。Silk Test Workbench は、記録するイメージ クリックごとに新しいイメージ資産を作成します。


 **注:** イメージ クリックの記録は、Java AWT/Swing コントロールを使用するアプリケーションまたはアプレットではサポートされません。

イメージ解決メソッド

Silk Test Workbench では、イメージ解決用に次のメソッドが用意されています。

メソッド	説明
ImageClick	資産で指定されたイメージの中央をクリックします。イメージが見つかるか、オブジェクト解決タイムアウト (同期オプションで定義可能) が経過するまで待機します。
ImageExists	資産で指定されたイメージが存在するかどうかを返します。

メソッド	説明
ImageRectangle	資産で指定されたイメージのオブジェクト相対矩形を返します。
ImageClickFile	ファイルで指定されたイメージをクリックします。
ImageExistsFile	ファイルで指定されたイメージが存在するかどうかを返します。
ImageRectangleFile	ファイルで指定されたイメージのオブジェクト相対矩形を返します。
VerifyAsset	検証資産を実行します。検証に合格しなかった場合、VerificationFailedException がスローされます。

 **注:** イメージ解決メソッドは、画面に表示されていないコントロールに対しては機能しません。たとえば、スクロールして画面外にあるイメージに対しては、イメージ解決を使用できません。

イメージ資産

イメージ資産は、次の場合に使用できます。

- オブジェクト解決で識別できない、高度にカスタマイズされたコントロールを含むテスト アプリケーションを簡単に操作する場合。座標ベースのクリックの代わりにイメージ クリックを使用し、指定されたイメージをクリックできます。
- テスト対象アプリケーションのグラフィカル オブジェクト (グラフなど) をテストする場合。

イメージ資産は、イメージと、Silk Test Workbench で資産を操作するために必要な追加情報で構成されます。

Silk Test Workbench では、イメージ資産用に次のメソッドが用意されています。

メソッド	説明
ImageClick	指定されたイメージ資産の中央をクリックします。イメージが見つかるか、オブジェクト解決タイムアウト (同期オプションで定義可能) が経過するまで待機します。
ImageExists	指定されたイメージ資産があるかどうかを返します。
ImageRectangle	指定されたイメージ資産のオブジェクト相対矩形を返します。



イメージ資産の作成

イメージ資産は、次のいずれかの方法で作成できます。

- 新しいイメージ資産を既存のビジュアル テストまたはスクリプトに挿入。
- 記録時。
- **アセット ブラウザ** を使用。


新しいイメージ資産をメニューから **アセット ブラウザ** 作成するには、以下のステップを実行します。

1. **表示 > アセット ブラウザ** を選択するか、**開始画面** の **入門ガイド > アセット ブラウザ** を選択します。
2. メニューで、**新規イメージ資産** をクリックします。
3. **資産の種類** リストで **イメージ資産** を右クリックし、**新規イメージ資産** を選択します。イメージ資産の UI が開きます。
4. 資産にイメージを追加する方法を選択します。
 - 既存のイメージを使用する場合は、**参照** をクリックし、イメージ ファイルを選択します。
 - テスト対象アプリケーションの UI から新しいイメージをキャプチャする場合は、**キャプチャ** を選択します。Web アプリケーションをテストする場合、**アプリケーションの選択** ウィンドウからイメージをキャプチャするブラウザーを選択できます。

- イメージをキャプチャする前にテスト対象アプリケーションのメニューを展開したい場合などに、5秒間マウスを静止した後に新しいイメージをキャプチャする場合は、**静止時にキャプチャ**を選択します。
5. 新しいイメージをキャプチャする場合は、キャプチャする画面領域を選択し、**選択範囲をキャプチャします** をクリックします。
 6. **名前** テキスト フィールドに、新しいイメージ資産のわかりやすい名前を入力します。
 **注:** 既存のイメージを使用する場合は、イメージ資産の名前はデフォルトで既存のイメージの名前になります。
 7. 省略可能：**検証** をクリックして、Silk Test Workbench が AUT の UI でイメージ資産を見つけることができるかどうかを確認します。
 Web アプリケーションをテストする場合、**アプリケーションの選択** ウィンドウからイメージをキャプチャするブラウザを選択できます。
 8. 省略可能：**クリック位置** チェック ボックスをオンにすると、イメージ資産上でクリックを実行する位置を選択できます。
 デフォルトの位置は、イメージの中央です。x フィールドと y フィールドに位置を入力するか、イメージ上の位置を選択します。
 9. **精度レベル** を指定します。
 精度レベルは、検証されるイメージがテスト対象アプリケーションのイメージと異なってもよい度合いを定義し、これを超えて異なっている場合、Silk Test Workbench はイメージが異なっていると判断します。これは、画面解像度が異なる複数のシステムまたはブラウザをテストする場合に役立ちます。誤検出を防ぐため、できるだけ精度レベルを高くすることを推奨します。デフォルトの精度レベル値は、オプションで変更できます。
 **注:** **精度レベル** を 5 未満に設定した場合、イメージの実際の色が比較で考慮されなくなります。イメージのグレースケール表現だけが比較されます。
 - 10 **保存** をクリックするとイメージ資産が Silk Test Workbench リポジトリに保存されます。**保存して閉じる** をクリックすると、さらにイメージ資産の UI が閉じます。

新しいイメージ資産が、**アセットブラウザ**に表示され、これを使用してイメージ クリックを実行できます。

同じイメージ資産に複数のイメージを追加できます。

-  **注:** モバイル ブラウザーに対する記録中にイメージ クリックを追加するには、**記録** ウィンドウで右クリックして、操作のリストから **ImageClick** を選択します。

同じイメージ資産に複数のイメージを追加する

テスト時に、異なるテスト構成を使用して、複数の環境の機能をテストする必要が生じることがよくあります。環境が異なると、イメージ資産にキャプチャしたイメージと実際のイメージが若干異なることがあり、イメージが存在するにもかかわらずイメージ クリックが失敗することがあります。このような場合、同じイメージ資産に複数のイメージを追加できます。

イメージ資産に別のイメージを追加するには、以下を実行します。

1. **表示** > **アセットブラウザ** を選択するか、**開始画面** の **入門ガイド** > **アセットブラウザ** を選択します。
2. **資産の種類** リストから **イメージ資産** を選択します。
3. イメージ資産に追加したいイメージをダブルクリックします。イメージ資産の UI が開きます。
4. UI の下部に表示されるプラス記号をクリックして、新しいイメージをイメージ資産に追加します。
5. **保存** をクリックするとイメージ資産が Silk Test Workbench リポジトリに保存されます。**保存して閉じる** をクリックすると、さらにイメージ資産の UI が閉じます。

新しいイメージが資産に追加されます。イメージ クリックが呼び出されるたびに、一致するものに到達するまで、Silk Test Workbench は資産のイメージとテスト対象アプリケーションの UI のイメージを比較します。デフォルトで、Silk Test Workbench は資産に追加された順にイメージを比較します。



注: Silk Test Workbench で比較するイメージの順番を変更するには、イメージ資産の UI の下部でイメージをクリックし、目的の場所にドラッグします。左から右の順に比較されます。最初に比較されるのは、最も左にあるイメージです。

スクリプトから資産を開く

スクリプトを編集しているときに、資産を右クリックして **資産を開く** を選択し、資産を開くことができます。これにより、GUI で資産が開きます。

資産がシステム上のファイルへの参照である場合 (ImageClickFile によって参照される場合など)、ファイルはシステムのデフォルト エディターで開かれます。

イメージクリックの検証資産名での変数の使用

ビジュアルテストからイメージクリックを実行するとき、ビジュアルテスト変数、アクティブデータ変数、式を使用して、イメージ資産の名前を指定できます。これにより、変数や式の値に応じて、さまざまなイメージ資産で同じイメージクリックを実行できます。

VB .NET スクリプトでは、ImageClick や ImageExists などのイメージ解決メソッドを介してこの機能をサポートします。これらのメソッドに対して、イメージ資産名を文字列で指定します。この文字列を、変数や式の名前とすることができます。

ビジュアルスクリプトの場合は、イメージクリックテストステップのプロパティでイメージ資産の名前を指定する必要があります。

1. イメージクリックが含まれているビジュアルテストを開きます。
2. イメージクリックテストステップを選択します。
3. **プロパティ** ペインを開きます。
4. **パラメータ** ノードを展開します。
5. **image** フィールドの ... をクリックし、イメージ資産名として使用する変数の名前を選択します。

イメージ検証

イメージ検証を使用して、テスト対象アプリケーション (AUT) の UI にイメージがあるかどうかをチェックできます。

イメージ検証は、イメージと、Silk Test Workbench で資産を操作するために必要な追加情報で構成されます。

イメージ検証を実行するには、VerifyAsset メソッドを使用します。

Silk Test Workbench が AUT のイメージを見つけることができなかった場合、イメージ検証は失敗します。この場合、スクリプトは続行します。

AUT 内でイメージ検証のロケータが見つからなかった場合、Silk Test Workbench は ObjectNotFoundException をスローします。

イメージ検証の作成

イメージ検証は、次のいずれかの方法で作成できます。

- **アセットブラウザ** を使用。
- 記録時。
- 新しいイメージ検証を既存のビジュアルテストに挿入。

新しいイメージ検証を **アセットブラウザ** で作成するには、以下のステップを実行します。

1. **表示** > **アセットブラウザ** を選択するか、**開始画面** の **入門ガイド** > **アセットブラウザ** を選択します。
2. **資産の種類** リストで **イメージ検証** を右クリックし、**新規イメージ検証** を選択します。イメージ検証の UI が開きます。
3. **識別** をクリックして、テスト対象アプリケーションの、検証するイメージを識別します。
4. **名前** テキスト フィールドに、新しいイメージ検証のわかりやすい名前を入力します。



注: 既存のイメージを使用する場合は、イメージ検証の名前はデフォルトで既存のイメージの名前になります。

5. 省略可能：最初にキャプチャしたイメージから変更されたために、テスト対象アプリケーションから同じイメージを再キャプチャする必要がある場合は、**再キャプチャ** をクリックします。

Web アプリケーションをテストする場合、**アプリケーションの選択** ウィンドウからイメージをキャプチャするブラウザを選択できます。

6. 省略可能：**検証** をクリックすると、イメージ検証が機能するかどうかをテストできます。Silk Test Workbench は、AUT の UI でイメージを、上から下へ、左から右へと検索し、最初に一致したイメージをハイライトします。
7. 省略可能：Silk Test Workbench がイメージ検証とテスト対象アプリケーション (AUT) の UI を比較するときに考慮しない除外領域をイメージ検証に追加できます。
8. 省略可能：オプション **クライアント領域のみ** を設定して、Silk Test Workbench がイメージ検証と AUT の UI を比較するときに、実際に AUT の一部であるイメージの部分だけを考慮するように定義できます。
9. **精度レベル** を指定します。

精度レベルは、検証されるイメージがテスト対象アプリケーションのイメージと異なってもよい度合いを定義し、これを超えて異なっている場合、Silk Test Workbench はイメージが異なっていると判断します。これは、画面解像度が異なる複数のシステムまたはブラウザをテストする場合に役立ちます。誤検出を防ぐため、できるだけ精度レベルを高くすることを推奨します。デフォルトの精度レベル値は、オプションで変更できます。



注: 精度レベル を 5 未満に設定した場合、イメージの実際の色が比較で考慮されなくなります。イメージのグレースケール表現だけが比較されます。

- 10 **保存** をクリックすると新しいイメージ検証が Silk Test Workbench リポジトリに保存されます。**保存して閉じる** をクリックすると、さらにイメージ検証の UI が閉じます。

新しいイメージ検証が**アセットブラウザ** に表示され、これを使用して、テスト対象アプリケーションの UI にイメージが存在するかどうかをチェックできます。

ビジュアル テストにイメージ検証を追加する

イメージ検証をビジュアル テストに追加して、テスト対象アプリケーションの UI に認識されないコントロールがあるかどうかをチェックできます。ビジュアル テストにイメージ検証を追加するには、以下を実行します。

1. ビジュアル テストを開きます。
2. イメージ検証を追加する場所の次のステップを右クリックします。
3. **挿入** > **テスト ロジック** > **検証** をクリックします。**テスト ロジック デザイナ** が開きます。
4. **次へ** をクリックします。**ロジックの種類**の**選択** ページが開きます。
5. **検証資産** をクリックします。**検証資産の構築** ページが開きます。
6. 次のいずれか 1 つのステップを行います：
 - イメージ検証の UI で新しいイメージ検証を作成するには、**新しい検証の作成** をクリックし、。
 - 既存のイメージ検証資産を挿入するには、**既存の検証の挿入** をクリックし、。
7. **次へ** をクリックします。**要約** ページが開き、新しい検証のプレビューが表示されます。
8. **終了** をクリックします。新しいイメージ検証が追加されます。

記録中にイメージ検証を追加する

イメージ検証をビジュアルテストまたは VB .NET スクリプトに追加して、テスト対象アプリケーションの UI に認識されないコントロールがあるかどうかをチェックできます。ビジュアルテストまたは VB .NET スクリプトの記録中にイメージ検証を追加するには、以下のステップを実行します。

1. 記録を開始します。
2. 検証するオブジェクトの上にマウスカーソルを移動して、**Ctrl+Alt** を押します。
モバイル Web アプリケーションを記録する場合は、オブジェクトをクリックして **検証の追加** をクリックすることもできます。
このオプションを実行すると、記録が一時的に停止され、**テスト ロジック デザイナ** ウィザードが表示されます。
3. **次へ** をクリックします。 **ロジックの種類を選択** ページが開きます。
4. **検証資産** をクリックします。 **検証資産の構築** ページが開きます。
5. 次のいずれか 1 つのステップを行います：
 - イメージ検証の UI で新しいイメージ検証を作成するには、**新しい検証の作成** をクリックし、。
 - 既存のイメージ検証資産を挿入するには、**既存の検証の挿入** をクリックし、。
6. **次へ** をクリックします。 **要約** ページが開き、新しい検証のプレビューが表示されます。
7. **終了** をクリックします。新しいイメージ検証が追加されます。
8. 記録を続けます。

イメージ検証が失敗した理由の確認

ビジュアルテストまたはスクリプトでイメージ検証の実行が失敗した場合、**イメージ検証の相違点ビュー** を開き、テスト対象アプリケーションで見つかった実際のイメージと予想されるイメージとの相違点を確認できます。

イメージ検証の相違点ビュー を開くには、以下のステップを実行します。

1. イメージ検証が失敗した実行の結果を開きます。
2. ビジュアルテストまたはスクリプトで失敗したイメージ検証のステップを右クリックします。
TrueLog Explorer **イメージ検証の相違点ビュー** が開きます。
3. 予想されるイメージと実際のイメージを比較し、イメージ検証が失敗した理由を確認します。
4. 省略可能：AUT の実際のイメージが変更されたためにイメージ検証が失敗している場合は、**予想として保存** をクリックして、実際のイメージを予想されるイメージに設定します。

イメージ検証の検証資産名での変数の使用

ビジュアルテストからイメージ検証を実行するとき、ビジュアルテスト変数、アクティブデータ変数、式を使用して、検証資産の名前を指定できます。これにより、変数や式の値に応じて、さまざまな検証資産で同じイメージ検証を実行できます。

VB .NET スクリプトでは、ImageClick や ImageExists などのイメージ解決メソッドを介してこの機能をサポートします。これらのメソッドに対して、イメージ資産名を文字列で指定します。この文字列を、変数や式の名前とすることができます。

ビジュアルスクリプトの場合は、イメージ検証テストステップのプロパティで検証資産の名前を指定する必要があります。

1. イメージ検証が含まれているビジュアルテストを開きます。
2. イメージ検証テストステップを選択します。
3. **プロパティ** ペインを開きます。

4. **名前** フィールドの ... をクリックし、検証資産名として使用する変数の名前を選択します。

イメージ検証の相違点ビュー

ビジュアルテストまたはスクリプトでイメージ検証の実行が失敗すると、**イメージ検証の相違点ビュー**に、テスト対象アプリケーションで見つかった実際のイメージと予想されるイメージとの相違点が表示されます。

イメージ検証の相違点ビューには、実際のイメージと予想されるイメージ、イメージ名、比較されたイメージのバージョンが表示されます。

特定の環境のテスト

Silk Test Workbench では、複数の種類の環境でのテストがサポートされています。


Apache Flex のサポート

Silk Test Workbench は、Internet Explorer やスタンドアロンの Flash Player を使用した Apache Flex アプリケーション、および Apache Flex 4 以降でビルドした Adobe AIR アプリケーションのテストを組み込みでサポートしています。

Silk Test Workbench では、Apache Flex 3.x および 4.x アプリケーションにおいて複数のアプリケーション ドメインもサポートされているため、サブアプリケーションをテストできます。Silk Test Workbench では、ロケーター階層ツリーの各サブアプリケーションが、関連するアプリケーション ドメイン コンテキストを持つアプリケーション ツリーとして認識されます。Apache Flex 4.x サブアプリケーションでは、ロケーター属性テーブルのルート レベルで SparkApplication クラスが使用されます。Apache Flex 3.x サブアプリケーションでは、FlexApplication クラスが使用されます。

サポートするコントロール

Apache Flex のテストで記録および再生できるコントロールの完全なリストについては、「Flex クラス リファレンス」を参照してください。

 **注:** Silk Test Flex オートメーション SDK は、Apache Flex のオートメーション API に基づいています。Silk Test オートメーション SDK は、Apache Flex のオートメーション API でサポートされているものと同じコンポーネントが同様にサポートされます。たとえば、Flex オートメーション API の typekey ステートメントでは、すべてのキーはサポートされません。テキスト入力ステートメントを使用してこの問題を解決できます。Flex オートメーション API の詳細については、『Apache Flex リリース ノート』を参照してください。

Adobe Flash Player で実行するための Flex アプリケーションの構成

Apache Flex アプリケーションを Flash Player で実行するには、以下のいずれか、または両方の条件が満たされている必要があります。

- Flex アプリケーションを作成する開発者は、アプリケーションを EXE ファイルとしてコンパイルする必要があります。アプリケーションは、ユーザーが起動すると、Flash Player で開きます。Windows Flash Player は、<http://www.adobe.com/support/flashplayer/downloads.html> からインストールします。
 - ユーザーが、Windows Flash Player Projector をインストールしている必要があります。ユーザーは、Flex の .SWF ファイルを開いた場合に Flash Player で開くように構成できます。Apache Flex 開発者スイートをインストールしないと、Flash Player をインストールしても Windows Flash Projector はインストールされません。Windows Flash Projector は、<http://www.adobe.com/support/flashplayer/downloads.html> からインストールします。
1. Microsoft Windows 7 および Microsoft Windows Server 2008 R2 では、管理者として実行されるように Flash Player を構成します。以下の手順を実行します。
 - a) Adobe Flash Player プログラム ショートカットまたは FlashPlayer.exe ファイルを右クリックして、**プロパティ** をクリックします。
 - b) **プロパティ** ダイアログ ボックスで、**互換性** タブをクリックします。

- c) **管理者としてこのプログラムを実行する** チェック ボックスをオンにして、**OK** をクリックします。
2. コマンド プロンプト (cmd.exe) で以下のコマンドを入力して、Flash Player で .SWF ファイルを起動します。

```
"<Application_Install_Directory>%ApplicationName.swf"
```

デフォルトで、<SilkTest_Install_Directory> は Program Files¥Silk¥Silk Test にあります。

Component Explorer の起動

Silk Test には、Component Explorer というサンプルの Apache Flex アプリケーションが含まれています。Component Explorer は、Adobe オートメーション SDK および Silk Test 固有のオートメーション実装を使用してコンパイルされており、テスト用に事前に構成されています。

Internet Explorer で、<http://demo.borland.com/flex/SilkTest20.0/index.html> を開きます。デフォルトブラウザでアプリケーションが起動します。

Apache Flex アプリケーションのテスト

Silk Test は、Apache Flex アプリケーションのテストを組み込みでサポートしています。Silk Test では、いくつかのサンプル Apache Flex アプリケーションを提供しています。サンプル アプリケーションには、<http://demo.borland.com/flex/SilkTest20.0/index.html> からアクセスできます。


新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

独自の Apache Flex アプリケーションをテストする前に、Apache Flex 開発者は以下のステップを実行する必要があります。

- Apache Flex アプリケーションのテストの有効化
- テスト可能な Apache Flex アプリケーションの作成
- Apache Flex コンテナのコーディング
- カスタム コントロールのオートメーション サポートの実装

独自の Apache Flex アプリケーションをテストするには、以下のステップを実行します。

- ローカルの Flash Player のセキュリティ設定の構成
- ビジュアル テストの記録
- ビジュアル テストの再生
- Apache Flex スクリプトのカスタマイズ
- カスタム Apache Flex コントロールのテスト

 **注:** Apache Flex アプリケーションを読み込み、Flex オートメーション フレームワークを初期化するとき、テストを実行するマシンおよび Apache Flex アプリケーションの複雑度に応じて、多少の時間がかかる場合があります。アプリケーションが完全に読み込まれるように、ウィンドウのタイムアウト値を高い値に設定します。

Apache Flex カスタム コントロールのテスト

Silk Test Workbench では、Apache Flex カスタム コントロールのテストがサポートされています。ただし、デフォルトでは、Silk Test Workbench は、カスタム コントロールの個別のサブコントロールを記録および再生することはできません。

カスタム コントロールをテストする場合、以下のオプションが存在します。

- 基本サポート

基本サポートでは、動的呼び出しを使用して、再生中にカスタム コントロールと対話します。作業量が少なく済むこのアプローチは、テスト アプリケーションにおいて、Silk Test Workbench が公開しないカスタム コントロールのプロパティおよびメソッドにアクセスする場合に使用します。カスタム コ

トロールの開発者は、コントロールのテストを容易にすることのみを目的としたメソッドおよびプロパティをカスタム コントロールに追加することもできます。ユーザーは、動的呼び出し機能を使用してこれらのメソッドやプロパティを呼び出すことができます。

基本サポートには以下のような利点があります。

- 動的呼び出しでは、テスト アプリケーションのコードを変更する必要がありません。
- 動的呼び出しを使用することによって、ほとんどのテストのニーズを満たすことができます。

基本サポートには以下のような短所があります。

- ロケーターには、具体的なクラス名が組み込まれません (たとえば、Silk Test Workbench では「//FlexSpinner」ではなく「//FlexBox」と記録されます)。
- 記録のサポートが限定されます。
- Silk Test Workbench では、イベントを再生できません。

例を含む動的呼び出しの詳細については、「*Apache Flex* メソッドの動的呼び出し」を参照してください。

- 高度なサポート

高度なサポートでは、カスタム コントロールに対して、特定のオートメーション サポートを作成できます。この追加のオートメーション サポートによって、記録のサポートおよびより強力な再生のサポートが提供されます。高度なサポートには以下のような利点があります。

- イベントの記録と再生を含む、高レベルの記録および再生のサポートが提供されます。
- Silk Test Workbench では、カスタム コントロールが他のすべての組み込み Apache Flex コントロールと同様に処理されます。
- Silk Test Workbench API とシームレスに統合できます。
- Silk Test Workbench では、ロケーターで具体的なクラス名が使用されます (たとえば、Silk Test Workbench では「//FlexSpinner」と記録されます)。

高度なサポートには以下のような短所があります。

- 実装作業が必要です。テスト アプリケーションを変更し、Open Agent を拡張する必要があります。

Flex メソッドの動的呼び出し

動的呼び出し機能を使用して Silk Test Workbench が対象としないコントロールのメソッドを呼び出したリ、プロパティを取得/設定することができます。この機能は、カスタム コントロールを使用したり、カスタマイズせずに Silk Test Workbench がサポートするコントロールを使用する場合に有効です。



注: 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。

オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。



注: 通常、ほとんどのプロパティは読み取り専用で、設定できません。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- Flex API で定義されているすべてのパブリック メソッド
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型

Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point など) が含まれます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

例

たとえば、アプリケーション開発者が次のメソッドとプロパティを持つ Calculator カスタム コントロールを作成したとします。

```
public function reset() : void
public function add(number1 : int, number2 : int) : int
public function get description : String
```

テスト担当者は、テスト内からメソッドを直接呼び出すことができます。例：

```
customControl.Invoke("reset")
Dim sum as Integer = customControl.Invoke("add", 1, 2)
Dim description As String = customControl.GetProperty("description")
```

テスト アプリケーションでのカスタム コントロールの定義

通常、テスト アプリケーションには、アプリケーションの開発中に追加されたカスタム コントロールがすでに含まれています。テスト アプリケーションにすでにカスタム コントロールが含まれている場合は、「動的呼び出しを使用して Flex カスタム コントロールをテストする」または「オートメーション サポートを使用してカスタム コントロールをテストする」に進んでください。

この手順では、Flex アプリケーション開発者が Flex で Spinner カスタム コントロールを作成する方法を示します。このトピックで作成する Spinner カスタム コントロールは、カスタム コントロールの実装およびテストのプロセスを説明するために、いくつかのトピックで使用されています。

Spinner カスタム コントロールは、以下のグラフィックに示すように、2 つのボタンと 1 つのテキスト フィールドを含んでいます。



ユーザーは、**Down** をクリックしてテキスト フィールドに表示されている値を 1 減分させ、**Up** をクリックしてテキスト フィールドの値を 1 増分させることができます。

カスタム コントロールには、設定および取得が可能なパブリックの Value プロパティが用意されています。

1. テスト アプリケーションで、コントロールのレイアウトを定義します。
たとえば、Spinner コントロール タイプでは、以下のように記述します。

```
<?xml version="1.0" encoding="utf-8"?>
<customcontrols:SpinnerClass xmlns:mx="http://www.adobe.com/2006/mxml"
xmlns:controls="mx.controls.*" xmlns:customcontrols="customcontrols.*">
  <controls:Button id="downButton" label="Down" />
  <controls:TextInput id="text" enabled="false" />
```

```
<controls:Button id="upButton" label="Up"/>
</customcontrols:SpinnerClass>
```

2. カスタム コントロールの実装を定義します。
たとえば、Spinner コントロール タイプでは、以下のように記述します。

```
package customcontrols
{
    import flash.events.MouseEvent;

    import mx.containers.HBox;
    import mx.controls.Button;
    import mx.controls.TextInput;
    import mx.core.UIComponent;
    import mx.events.FlexEvent;

    [Event(name="increment",      type="customcontrols.SpinnerEvent")]
    [Event(name="decrement",     type="customcontrols.SpinnerEvent")]

    public class SpinnerClass extends HBox
    {
        public var downButton : Button;
        public var upButton : Button;
        public var text : TextInput;
        public var ssss: SpinnerAutomationDelegate;
        private var _lowerBound : int = 0;
        private var _upperBound : int = 5;

        private var _value : int = 0;
        private var _stepSize : int = 1;

        public function SpinnerClass() {
            addEventListener(FlexEvent.CREATION_COMPLETE,
creationCompleteHandler);
        }

        private function creationCompleteHandler(event:FlexEvent) : void {
            downButton.addEventListener(MouseEvent.CLICK, downButtonClickHandler);
            upButton.addEventListener(MouseEvent.CLICK, upButtonClickHandler);
            updateText();
        }

        private function downButtonClickHandler(event : MouseEvent) : void {
            if(Value - stepSize >= lowerBound) {
                Value = Value - stepSize;
            }
            else {
                Value = upperBound - stepSize + Value - lowerBound + 1;
            }

            var spinnerEvent : SpinnerEvent = new
SpinnerEvent(SpinnerEvent.DECREMENT);
            spinnerEvent.steps = _stepSize;
            dispatchEvent(spinnerEvent);
        }

        private function upButtonClickHandler(event : MouseEvent) : void {
            if(cValue <= upperBound - stepSize) {
                Value = Value + stepSize;
            }
        }
    }
}
```

```

    }
    else {
        Value = lowerBound + Value + stepSize - upperBound - 1;
    }

    var spinnerEvent : SpinnerEvent = new
SpinnerEvent(SpinnerEvent.INCREMENT);
    spinnerEvent.steps = _stepSize;
    dispatchEvent(spinnerEvent);
}

private function updateText() : void {
    if(text != null) {
        text.text = _value.toString();
    }
}

public function get Value() : int {
    return _value;
}

public function set Value(v : int) : void {
    _value = v;
    if(v < lowerBound) {
        _value = lowerBound;
    }
    else if(v > upperBound) {
        _value = upperBound;
    }
    updateText();
}

public function get stepSize() : int {
    return _stepSize;
}

public function set stepSize(v : int) : void {
    _stepSize = v;
}

public function get lowerBound() : int {
    return _lowerBound;
}

public function set lowerBound(v : int) : void {
    _lowerBound = v;
    if(Value < lowerBound) {
        Value = lowerBound;
    }
}

public function get upperBound() : int {
    return _upperBound;
}

public function set upperBound(v : int) : void {
    _upperBound = v;
    if(Value > upperBound) {

```

```

        Value = upperBound;
    }
}
}
}
}

```

3. コントロールが使用するイベントを定義します。
たとえば、Spinner コントロール タイプでは、以下のように記述します。

```

package customcontrols
{
    import flash.events.Event;

    public class SpinnerEvent extends Event
    {
        public static const INCREMENT : String = "increment";
        public static const DECREMENT : String = "decrement";

        private var _steps : int;

        public function SpinnerEvent(eventName : String) {
            super(eventName);
        }

        public function set steps(value:int) : void {
            _steps = value;
        }


        public function get steps() : int {
            return _steps;
        }
    }
}

```

次のステップでは、テスト アプリケーションのオートメーション サポートを実装します。

動的呼び出しを使用した Flex カスタム コントロールのテスト

Silk Test Workbench では、動的呼び出しを使用したカスタム コントロールの記録と再生のサポートが提供されており、これにより再生中にカスタム コントロールを操作できます。作業量が少なく済むこのアプローチは、テスト アプリケーションにおいて、Silk Test Workbench が公開しないカスタム コントロールのプロパティおよびメソッドにアクセスする場合に使用します。カスタム コントロールの開発者は、コントロールのテストを容易にすることのみを目的としたメソッドおよびプロパティをカスタム コントロールに追加することもできます。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。

1. コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。
2. オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。
3. オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。
4. オブジェクトの複数の動的メソッドは invokeMethods メソッドを使用して呼び出します。
5. コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。
6. 動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。

例

この例では、以下の図に示すように、2つのボタンと1つのテキストフィールドを含む Spinner カスタム コントロールをテストします。



ユーザーは、**Down** をクリックしてテキスト フィールドに表示されている値を 1 減分させ、**Up** をクリックしてテキスト フィールドの値を 1 増分させることができます。

カスタム コントロールには、設定および取得が可能なパブリックの Value プロパティが用意されています。

Spinner の値を 4 に設定するには、以下のように入力します。

```
Dim spinner = Desktop.Find("//  
FlexBox[@className=customcontrols.Spinner]")  
spinner.SetProperty("Value", 4)
```

オートメーション サポートを使用したカスタム コントロールのテスト

カスタム コントロールに対して、特定のオートメーション サポートを作成できます。この追加のオートメーション サポートによって、記録のサポートおよびより強力な再生のサポートが提供されます。オートメーション サポートを作成するには、テスト アプリケーションを変更し、Open Agent を拡張する必要があります。

Silk Test Workbench でカスタム コントロールをテストする前に、以下のステップを実行します。

- テスト アプリケーションでのカスタム コントロールの定義
- オートメーション サポートの実装

テスト アプリケーションを変更してオートメーション サポートを組み込んだあと、以下のステップを実行します。

1. ビジュアル テストでは、追加の構成は必要ありません。必要に応じて、カスタム コントロールに対してビジュアル テストを記録および再生できます。
2. スクリプトの場合は、スクリプトを記録して、カスタム コントロールに合わせて手動で変更を加えます。
たとえば、以下のコードでは、オートメーションの委譲に実装されている「Increment」メソッドを使用して、Spinner の値を 3 増分する方法を示しています。

```
_desktop.TestObject("//FlexSpinner[@caption='index:1']").Invoke("Increment", 3)
```

以下の例は、Spinner の値を 3 に設定する方法を示しています。

```
_desktop.TestObject("//FlexSpinner[@caption='index:1']").SetProperty("Value", 3)
```

カスタム コントロールのオートメーション サポートの実装

カスタム コントロールをテストする前に、カスタム コントロールの ActionScript でオートメーション サポート（オートメーションの委譲）を実装し、テスト アプリケーションにコンパイルします。

以下の手順では、Flex のカスタム Spinner コントロールを使用して、カスタム コントロールのオートメーション サポートの実装方法を示します。Spinner カスタム コントロールは、以下のグラフィックに示すように、2つのボタンと1つのテキストフィールドを含んでいます。



ユーザーは、**Down** をクリックしてテキスト フィールドに表示されている値を 1 減分させ、**Up** をクリックしてテキスト フィールドの値を 1 増分させることができます。

カスタム コントロールには、設定および取得が可能なパブリックの Value プロパティが用意されています。

1. カスタム コントロールの ActionScript でオートメーション サポート（オートメーションの委譲）を実装します。

オートメーションの委譲の実装の詳細については、Adobe Live ドキュメント (http://livedocs.adobe.com/flex/3/html/help.html?content=functest_components2_14.html) を参照してください。

この例では、オートメーションの委譲によって、「increment」および「decrement」メソッドに対してサポートが追加されます。オートメーションの委譲のコード例は以下のとおりです。

```
package customcontrols
{
    import flash.display.DisplayObject;
    import mx.automation.Automation;
    import customcontrols.SpinnerEvent;
    import mx.automation.delegates.containers.BoxAutomationImpl;
    import flash.events.Event;
    import mx.automation.IAutomationObjectHelper;
    import mx.events.FlexEvent;
    import flash.events.IEventDispatcher;
    import mx.preloaders.DownloadProgressBar;
    import flash.events.MouseEvent;
    import mx.core.EventPriority;

    [Mixin]
    public class SpinnerAutomationDelegate extends BoxAutomationImpl
    {

        public static function init(root:DisplayObject) : void {
            // register delegate for the automation
            Automation.registerDelegateClass(Spinner, SpinnerAutomationDelegate);
        }

        public function SpinnerAutomationDelegate(obj:Spinner) {
            super(obj);
            // listen to the events of interest (for recording)
            obj.addEventListener(SpinnerEvent.DECREMENT, decrementHandler);
            obj.addEventListener(SpinnerEvent.INCREMENT, incrementHandler);
        }

        protected function decrementHandler(event : SpinnerEvent) : void {
            recordAutomatableEvent(event);
        }

        protected function incrementHandler(event : SpinnerEvent) : void {
            recordAutomatableEvent(event);
        }

        protected function get spinner() : Spinner {
            return uiComponent as Spinner;
        }

        //-----
        // override functions
    }
}
```

```

//-----
override public function get automationValue():Array {
    return [ spinner.Value.toString() ];
}

private function replayClicks(button : IEventDispatcher, steps : int) : Boolean
{
    var helper : IAutomationObjectHelper =
Automation.automationObjectHelper;
    var result : Boolean;
    for(var i:int; i < steps; i++) {
        helper.replayClick(button);
    }
    return result;
}

override public function replayAutomatableEvent(event:Event):Boolean {

    if(event is SpinnerEvent) {
        var spinnerEvent : SpinnerEvent = event as SpinnerEvent;
        if(event.type == SpinnerEvent.INCREMENT) {
            return replayClicks(spinner.upButton, spinnerEvent.steps);
        }
        else if(event.type == SpinnerEvent.DECREMENT) {
            return replayClicks(spinner.downButton, spinnerEvent.steps);
        }
        else {
            return false;
        }
    }
    else {
        return super.replayAutomatableEvent(event);
    }
}

// do not expose the child controls (i.e the buttons and the textfield) as
individual controls
override public function get numAutomationChildren():int {
    return 0;
}
}
}

```

2. Open Agent にオートメーションの委譲を導入するために、カスタム コントロールを記述する XML ファイルを作成します。

クラス定義ファイルには、インストルメント化されたすべての Flex コンポーネントについての情報が含まれています。このファイルでは、記録中にイベントを送信でき、再生中にイベントを受け取ることができるコンポーネントについての情報が提供されます。クラス定義ファイルには、サポートされているプロパティの定義も含まれています。

Spinner カスタム コントロールの XML ファイルは以下のようになります。

```

<?xml version="1.0" encoding="UTF-8"?>
<TypeInfoInformation>
    <ClassInfo Name="FlexSpinner" Extends="FlexBox">
        <Implementation
            Class="customcontrols.Spinner" />
        <Events>

```

```

    <Event Name="Decrement">
      <Implementation
        Class="customcontrols.SpinnerEvent"
        Type="decrement" />
      <Property Name="steps">
        <PropertyType Type="integer" />
      </Property>
    </Event>
    <Event Name="Increment">
      <Implementation
        Class="customcontrols.SpinnerEvent"
        Type="increment" />
      <Property Name="steps">
        <PropertyType Type="integer" />
      </Property>
    </Event>
  </Events>
  <Properties>
    <Property Name="lowerBound" accessType="read">
      <PropertyType Type="integer" />
    </Property>
    <Property Name="upperBound" accessType="read">
      <PropertyType Type="integer" />
    </Property>
    <!-- expose read and write access for the Value property -->
    <Property Name="Value" accessType="both">
      <PropertyType Type="integer" />
    </Property>
    <Property Name="stepSize" accessType="read">
      <PropertyType Type="integer" />
    </Property>
  </Properties>
</ClassInfo>
</TypeInfo>

```

3. サポートされている Flex コントロールのすべてのクラス、およびそのメソッドとプロパティを記述するすべての XML ファイルが格納されるフォルダに、カスタムコントロールの XML ファイルを配置します。

Silk Test には、サポートされている Flex コントロールのすべてのクラス、およびそのメソッドとプロパティを記述するいくつかの XML ファイルが含まれています。これらの XML ファイルは、<Silk Test_install_directory>%ng%agent%plugins%com.borland.fastxd.techdomain.flex.agent_<バージョン>%config%automationEnvironment フォルダにあります。

独自の XML ファイルを提供する場合は、XML ファイルをこのフォルダにコピーする必要があります。Open Agent が起動して、Apache Flex のサポートを初期化する場合、このディレクトリの内容が読み込まれます。

Flex の Spinner サンプルコントロールをテストするには、CustomControls.xml ファイルをこのフォルダにコピーする必要があります。Open Agent が現在実行されている場合は、ファイルをフォルダにコピーしたあと、Open Agent を再起動します。

これで、カスタムコントロールを含むビジュアル テストまたはスクリプトを記録および再生できるようになりました。

Flex クラス定義ファイル

クラス定義ファイルには、インストール化されたすべての Flex コンポーネントについての情報が含まれています。このファイルでは、記録中にイベントを送信でき、再生中にイベントを受け取ることができるコンポーネントについての情報が提供されます。クラス定義ファイルには、サポートされているプロパティの定義も含まれています。

Silk Test には、Flex の共通コントロールおよび特殊化されたコントロールのすべてのクラス、イベント、およびプロパティを記述するいくつかの XML ファイルが含まれています。これらの XML ファイルは、`<Silk_Test_install_directory>%ng%agent%plugins%com.borland.fastxd.techdomain.flex.agent_<バージョン>%config%automationEnvironment` フォルダにあります。

独自の XML ファイルを提供する場合は、XML ファイルをこのフォルダにコピーする必要があります。Silk Test のエージェントが起動して Apache Flex のサポートを初期化するとき、このディレクトリの内容が読み込まれます。

XML ファイルの基本的な構造は以下のとおりです。

```
<TypeInfoInformation>
<ClassInfo>
<Implementation />
<Events>
<Event />
...
</Events>
<Properties>
<Property />
...
</Properties>
</ClassInfo>
</TypeInfoInformation>
```

Apache Flex スクリプトのカスタマイズ

手動で Flex スクリプトをカスタマイズできます。Flex オブジェクトのプロパティに対して Verify 関数を使用して、手動で検証を挿入できます。各 Flex オブジェクトには、検証可能な一連のプロパティがあります。検証に使用できるプロパティのリストについては、「Flex クラス リファレンス」を参照してください。


1. Flex アプリケーションのテストを記録します。
2. カスタマイズするスクリプト ファイルを開きます。
3. 追加するコードを手動で入力します。

同一 Web ページ上の複数の Flex アプリケーションのテスト


同じ Web ページに複数の Flex アプリケーションが存在する場合、Silk Test Workbench は、Flex アプリケーションの ID またはアプリケーションの size プロパティを使用して、テスト対象アプリケーションを特定します。同じページに複数のアプリケーションが存在し、それらのサイズが異なる場合、Silk Test Workbench は、size プロパティを使用して操作実行対象のアプリケーションを特定します。追加の操作は必要ありません。

以下の場合、Silk Test Workbench は、JavaScript を使用して Flex アプリケーションの ID を検索し、操作実行対象のアプリケーションを特定します。

- 単一の Web ページ上に複数の Flex アプリケーションが存在する場合。
- これらのアプリケーションのサイズが同じである場合。

 **注:** この場合、ブラウザ マシンで JavaScript が有効になっていないと、スクリプト実行時にエラーが発生します。

1. JavaScript を有効にします。
2. Internet Explorer で、以下の手順を実行します。
 - a) ツール > インターネット オプション を選択します。
 - b) セキュリティ タブをクリックします。
 - c) レベルのカスタマイズ をクリックします。
 - d) スクリプト作成 セクションの アクティブ スクリプト で、有効にする をクリックして OK をクリックします。
3. 「Apache Flex アプリケーションのテスト」 の手順に従います。

 **注:** Web ページにフレームが存在し、アプリケーションが同じサイズである場合、この方法は動作しません。

Adobe AIR のサポート

Silk Test Workbench がサポートする Adobe AIR でのテストは、Flex 4 コンパイラを使用してコンパイルされたアプリケーションのみです。サポートされているバージョンの詳細については、リリース ノートで最新の情報を確認してください。

Silk Test には、サンプルの Adobe AIR アプリケーションが含まれています。 <http://demo.borland.com/flex/SilkTest20.0/index.html> にあるサンプル アプリケーションにアクセスして、使用する Adobe AIR アプリケーションをクリックしてください。オートメーションあり、またはオートメーションなしのアプリケーションを選択できます。AIR アプリケーションを実行するには、Adobe AIR ランタイムをインストールする必要があります。

名前またはインデックスを使用する Flex の Select メソッドの概要

Flex の Select メソッドは、選択するコントロールの Name または Index を使用して記録できます。デフォルトで、Silk Test Workbench では、コントロールの名前を使用して Select メソッドが記録されます。ただし、コントロールのインデックスを使用して Select イベントを記録するように環境を変更したり、名前を使用した記録とインデックスを使用した記録を切り替えたりすることができます。

以下のコントロールでは、インデックスを使用して Select イベントを記録できます。


- FlexList
- FlexTree
- FlexDataGrid
- FlexAdvancedDataGrid
- FlexOLAPDataGrid
- FlexComboBox

デフォルト設定は、コントロールの名前を使用する ItemBasedSelection (Select イベント) です。インデックスを使用するには、IndexBasedSelection (SelectIndex イベント) を使用するように AutomationEnvironment を変更する必要があります。これらのクラスのいずれかの動作を変更するには、以下のコードを使用して FlexCommonControls.xml、AdvancedDataGrid.xml、または OLAPDataGrid.xml ファイルを変更する必要があります。これらの XML ファイルは、<Silk Test_install_directory>%ng%agent%plugins%com.borland.fastxd.techdomain.flex.agent_<バージョン>%config%automationEnvironment フォルダにあります。対応する xml ファイルで、以下の変更を行います。

```
<ClassInfo Extends="FlexList" Name="FlexControlName"
EnableIndexBasedSelection="true" >
```

```
...  
</ClassInfo>
```

この変更では、FlexList::SelectIndex イベントの記録に IndexBasedSelection が使用されています。コードの EnableIndexBasedSelection= を false に設定するか、またはこのブール値を削除すると、記録で名前が使用される設定に戻ります (FlexList::Select イベント)。

 **注:** これらの変更内容を有効にするには、アプリケーションを再起動する必要があります。アプリケーションを再起動すると、Silk Test Agent も自動的に再起動されます。

FlexDataGrid コントロールでの項目の選択

FlexDataGrid コントロールの項目は、インデックス値または内容値を使用して選択します。

1. インデックス値を使用して FlexDataGrid コントロールの項目を選択するには、SelectIndex メソッドを使用します。
たとえば、FlexDataGrid.SelectIndex(1) のように入力します。
2. 内容値を使用して FlexDataGrid コントロールの項目を選択するには、Select メソッドを使用します。
必要な形式の文字列を使用して、選択する行を識別します。項目と項目の間は、縦線文字 (|) で区切る必要があります。少なくとも 1 つの項目を 2 つのアスタリスク (*) で囲む必要があります。これにより、クリックが実行される項目が識別されます。
構文は FlexDataGrid.Select("*Item1* | Item2 | Item3") です。

Flex アプリケーションのテストの有効化


Flex アプリケーションをテストに対して有効化するには、Apache Flex 開発者は Flex アプリケーションに以下のコンポーネントを組み込む必要があります。

- Apache Flex オートメーション パッケージ
- Silk Test オートメーション パッケージ

Apache Flex オートメーション パッケージ

開発者は、Flex オートメーション パッケージを使用して、オートメーション API を使用する Flex アプリケーションを作成できます。Flex オートメーション パッケージは、Adobe の Web サイト (<http://www.adobe.com>) からダウンロードできます。パッケージには、以下の内容が含まれています。

- オートメーション ライブラリ : automation.swc ライブラリおよび automation_agent.swc ライブラリは、Flex フレームワーク コンポーネントの委譲の実装です。automation_agent.swc ファイルおよび関連するリソースバンドルは、汎用的なエージェント メカニズムです。Silk Test Agent などのエージェントは、これらのライブラリの上に構築されます。
- サンプル

 **注:** Silk Test Flex オートメーション SDK は、Flex のオートメーション API に基づいています。Silk Test オートメーション SDK は、Flex のオートメーション API でサポートされているものと同じコンポーネントが同様にサポートされます。たとえば、Flex オートメーション API の typekey ステートメントでは、すべてのキーはサポートされません。テキスト入カステートメントを使用してこの問題を解決できます。Flex オートメーション API の詳細については、『Apache Flex リリース ノート』を参照してください。

Silk Test オートメーション パッケージ

Silk Test の Open Agent は、Apache Flex オートメーション エージェント ライブラリを使用しています。FlexTechDomain.swc ファイルに、Silk Test 固有の実装が含まれています。

以下のいずれかの方法を使用して、アプリケーションをテストに対して有効化できます。

- Flex アプリケーションへのオートメーション パッケージのリンク
- 実行時の読み込み

Flex アプリケーションへのオートメーション パッケージのリンク

テストする予定の Flex アプリケーションを事前にコンパイルする必要があります。機能テスト クラスは、コンパイル時にアプリケーションに埋め込まれ、アプリケーションは実行時に自動テストに関する外部依存関係を持ちません。


コンパイル時にアプリケーションの SWF ファイルに機能テスト クラスを埋め込むと、SWF ファイルのサイズが大きくなります。SWF ファイルのサイズが重要でない場合は、機能テストと展開に同じ SWF ファイルを使用します。SWF ファイルのサイズが重要である場合は、2 つの SWF ファイルを生成します。1 つは機能テスト クラスが埋め込まれたファイル、もう 1 つは機能テスト クラスが埋め込まれていないファイルです。展開には、テスト クラスが埋め込まれていない SWF ファイルを使用します。

include-libraries コンパイラ オプションを指定してテストのために Flex アプリケーションを事前にコンパイルする場合は、以下のファイルを参照します。

- automation.swc
- automation_agent.swc
- FlexTechDomain.swc
- automation_charts.swc (アプリケーションでグラフおよび Flex 2.0 を使用する場合のみインクルード)
- automation_dmv.swc (アプリケーションでグラフおよび Flex 3.x 以降を使用する場合にインクルード)
- automation_flasflexkit.swc (アプリケーションで埋め込みの Flash コンテンツを使用する場合にインクルード)
- automation_spark.swc (アプリケーションで新しい Flex 4.x コントロールを使用する場合にインクルード)
- automation_air.swc (アプリケーションが AIR アプリケーションである場合にインクルード)
- automation_airspace.swc (アプリケーションが AIR アプリケーションであり、新しい Flex 4.x コントロールを使用する場合にインクルード)

Flex アプリケーションの最終リリース バージョンを作成する場合は、これらの SWC ファイルへの参照なしでアプリケーションを再コンパイルします。オートメーション SWC ファイルの使用の詳細については、『*Apache Flex リリース ノート*』を参照してください。

アプリケーションをサーバーに展開しないで、ファイル プロトコルを使用して要求したり、Apache Flex Builder 内で実行したりする場合は、各 SWF ファイルをローカルの信頼済みサンドボックスに組み込む必要があります。このためには、追加の構成情報が必要です。コンパイラの構成ファイルを変更するか、またはコマンド ライン オプションを使用して、構成情報を追加します。


 **注:** Silk Test Flex オートメーション SDK は、Flex のオートメーション API に基づいています。Silk Test オートメーション SDK は、Flex のオートメーション API でサポートされているものと同じコンポーネントが同様にサポートされます。たとえば、オートメーション コードを使用してアプリケーションがコンパイルされ、連続的に SWF ファイルが読み込まれる場合、メモリ リークが発生して、最終的にアプリケーションでメモリが不足します。Flex Control Explorer サンプル アプリケーションは、この問題の影響を受けます。回避策として、Explorer が読み込むアプリケーションの SWF ファイルをオートメーション ライブラリを使用してコンパイルしない方法があります。たとえば、Explorer のメイン アプリケーションのみをオートメーション ライブラリを使用してコンパイルします。SWFLoader の代わりにモジュール ローダーを使用する方法もあります。Flex オートメーション API の詳細については、『*Apache Flex リリース ノート*』を参照してください。

テストのための Flex アプリケーションの事前コンパイル

アプリケーションをテスト用に事前コンパイルするか、または実行時の読み込みを使用することによって、アプリケーションをテストに対して有効化できます。

1. 以下のコードを構成ファイルに追加することによって、コンパイラの構成ファイルに automation.swc、automation_agent.swc、および FlexTechDomain.swc ライブラリをインクルードします。

```
<include-libraries>
...
<library>/libs/automation.swc</library>
<library>/libs/automation_agent.swc</library>
<library>pathinfo/FlexTechDomain.swc</library>
</include-libraries>
```

 **注:** アプリケーションでグラフを使用する場合は、automation_charts.swc ファイルも追加する必要があります。


2. コマンドラインコンパイラで include-libraries コンパイラ オプションを使用して、automation.swc、automation_agent.swc、および FlexTechDomain.swc ライブラリの場所を指定します。構成ファイルは以下の場所にあります。


Apache Flex 2 SDK – <flex_installation_directory>/frameworks/flex-config.xml

Apache Flex データ サービス – <flex_installation_directory>/flex/WEB-INF/flex/flex-config.xml

以下の例では、automation.swc ファイルと automation_agent.swc ファイルがアプリケーションに追加されています。

```
mxmclc -include-libraries+=../frameworks/libs/automation.swc;../frameworks/libs/
automation_agent.swc;pathinfo/FlexTechDomain.swc MyApp.mxml
```

 **注:** コマンドラインで include-libraries オプションを明示的に設定すると、既存のライブラリに対して追加されるのではなく、既存のライブラリが上書きされます。コマンドラインで include-libraries オプションを使用して automation.swc ファイルと automation_agent.swc ファイルを追加する場合は、+= 演算子を使用します。これにより、インクルードされる既存のライブラリが上書きされるのではなく、インクルードされる既存のライブラリに対して追加されます。

 **注:** Silk Test Flex オートメーション SDK は、Flex のオートメーション API に基づいています。Silk Test オートメーション SDK は、Flex のオートメーション API でサポートされているものと同じコンポーネントが同様にサポートされます。たとえば、オートメーションコードを使用してアプリケーションがコンパイルされ、連続的に SWF ファイルが読み込まれる場合、メモリリークが発生して、最終的にアプリケーションでメモリが不足します。Flex Control Explorer サンプルアプリケーションは、この問題の影響を受けます。回避策として、Explorer が読み込むアプリケーションの SWF ファイルをオートメーションライブラリを使用してコンパイルしない方法があります。たとえば、Explorer のメインアプリケーションのみをオートメーションライブラリを使用してコンパイルします。SWFLoader の代わりにモジュールローダーを使用する方法もあります。Flex オートメーション API の詳細については、『Apache Flex リリースノート』を参照してください。

実行時の読み込み

Silk Test Flex オートメーションランチャを使用して、実行時に Flex オートメーションサポートを読み込むことができます。このアプリケーションは、オートメーションライブラリを使用してコンパイルされており、SWFLoader クラスを使用してユーザーのアプリケーションを読み込みます。これにより、SWF ファイルにオートメーションライブラリをコンパイルしなくても、アプリケーションが自動的にテストに対して有効化されます。Silk Test Flex オートメーションランチャは、HTML および SWF のファイル形式で利用できます。

制限事項

- Flex オートメーションランチャアプリケーションは、自動的にルートアプリケーションとなります。ユーザーのアプリケーションをルートアプリケーションにする必要がある場合は、Silk Test Flex オートメーションランチャを使用してオートメーションサポートを読み込むことができません。
- 外部ライブラリを読み込むアプリケーション（他の SWF ファイルライブラリを読み込むアプリケーション）をテストするには、自動テストに特別な設定が必要です。実行時に読み込まれるライブラリ（ランタイム共有ライブラリ（RSL）を含む）は、読み込むアプリケーションの ApplicationDomain に読み込まれる必要があります。アプリケーションで使用される SWF ファイルが異なるアプリケーションドメインに読み込まれた場合、自動テストの記録と再生が正しく動作しません。以下に、同じ ApplicationDomain に読み込まれるライブラリの例を示します。

```
import flash.display.*;

import flash.net.URLRequest;

import flash.system.ApplicationDomain;

import flash.system.LoaderContext;

var ldr:Loader = new Loader();

var urlReq:URLRequest = new URLRequest("RuntimeClasses.swf");

var context:LoaderContext = new LoaderContext();

context.applicationDomain = ApplicationDomain.currentDomain;

loader.load(request, context);
```

実行時読み込み

1. Silk Test Automation SDK Flex <バージョン> FlexAutomationLauncher ディレクトリの内容を、テストする Flex アプリケーションのディレクトリにコピーします。
2. Windows Explorer で FlexAutomationLauncher.html を開き、ファイルパスへの接尾辞として以下のパラメータを追加します。

```
?automationurl=YourApplication.swf
```

YourApplication.swf は Flex アプリケーションの SWF ファイルの名前です。

3. ファイルパスへの接頭辞として file:/// を追加します。
たとえば、ファイルの URL に ?automationurl=explorer.swf などのパラメータが含まれている場合は、以下のように入力します。を設定して、インストールするコンポーネントを指定できます。

```
file:///C:/Program%20Files/Silk/Silk Test/ng/sampleapplications/Flex/3.2/
FlexControlExplorer32/FlexAutomationLauncher.html?automationurl=explorer.swf
```

コマンドラインを使用した構成情報の追加

コマンドラインコンパイラを使用して automation.swc、automation_agent.swc、および FlexTechDomain.swc ライブラリの場所を指定するには、include-libraries コンパイラオプションを使用します。

次の例では、automation.swc ファイルと automation_agent.swc ファイルがアプリケーションに追加されます。

```
mxmmlc -include-libraries+=../frameworks/libs/automation.swc;../frameworks/libs/automation_agent.swc;pathinfo/FlexTechDomain.swc MyApp.mxml
```



注: アプリケーションでグラフを使用する場合は、include-libraries コンパイラ オプションに automation_charts.swc ファイルも追加する必要があります。

コマンドラインで include-libraries オプションを明示的に設定すると、既存のライブラリに対して追加されるのではなく、既存のライブラリが上書きされます。コマンドラインで include-libraries オプションを使用して automation.swc ファイルと automation_agent.swc ファイルを追加する場合は、+= 演算子を使用します。これにより、インクルードされる既存のライブラリが上書きされるのではなく、インクルードされる既存のライブラリに対して追加されます。

Flex Builder プロジェクトに自動テストサポートを追加するには、include-libraries コンパイラ オプションに automation.swc および automation_agent.swc ライブラリも追加する必要があります。

Flex アプリケーションにパラメータを渡す

以下の手順に従って、Flex アプリケーションにパラメータを渡すことができます。

実行する前に Flex アプリケーションにパラメータを渡す

オートメーション ライブラリを使用して、実行する前に Flex アプリケーションにパラメータを渡すことができます。

1. 適切なオートメーション ライブラリを使用して、アプリケーションをコンパイルします。
2. パラメータの指定には、通常どおり標準的な Flex のメカニズムを使用します。

Flex オートメーションランチャを使用して、実行時に Flex アプリケーションにパラメータを渡す

このタスクを開始する前に、実行時の読み込みに対応するようにアプリケーションを準備します。

1. FlexAutomationLauncher.html ファイルを開くか、または例として FlexAutomationLauncher.html を使用してファイルを作成します。
2. 以下のセクションに移動します。

```
<script language="JavaScript" type="text/javascript">
    AC_FL_RunContent(eef
        "src", "FlexAutomationLauncher",
        "width", "100%",
        "height", "100%",
        "align", "middle",
        "id", "FlexAutomationLauncher",
        "quality", "high",
        "bgcolor", "white",
        "name", "FlexAutomationLauncher",
        "allowScriptAccess","sameDomain",
        "type", "application/x-shockwave-flash",
```

```
        "pluginspage", "http://www.adobe.com/go/getflashplayer",
        "flashvars", "yourParameter=yourParameterValue"+
"&automationurl=YourApplication.swf"

    );
</script>
```



注: 「src」、 「id」、 および 「name」 の 「FlexAutomationLauncher」 の値は変更しないでください。

3. 「yourParameter=yourParameterValue」 に、独自のパラメータを追加します。
4. 「& automationurl=YourApplication.swf」 の値として、テストする Flex アプリケーションの名前を渡します。
5. ファイルを保存します。

テスト可能な Flex アプリケーションの作成

Flex 開発者は、Flex アプリケーションを可能なかぎりテストしやすくするためのテクニックを利用できます。以下のテクニックがあります。

- オブジェクトに対するわかりやすい ID の指定
- オブジェクトの重複の回避

オブジェクトに対するわかりやすい ID の指定

テストしやすいアプリケーションを作成するには、スクリプト内でオブジェクトを識別しやすくする必要があります。テストするすべてのコントロールに対して、わかりやすい文字列を使用した ID プロパティの値を設定できます。

オブジェクトに対してわかりやすい ID を指定するには：

- テスト可能なすべての MXML コンポーネントに対して ID を指定して、その Flex コントロールの参照時にテスト スクリプトで一意的 ID が使用できるようにします。
- これらの ID は、ユーザーがテスト スクリプト内でそのオブジェクトを容易に識別できるように、可能なかぎり人間が理解しやすい文字列にします。たとえば、TabNavigator 内の Panel コンテナの id プロパティは、panel1 や p1 ではなく submit_panel とします。

Silk Test Workbench を使用する場合、id や childIndex などの特定のタグに基づいて、オブジェクトに対して自動的に名前が設定されます。id プロパティに値がない場合、Silk Test Workbench では、childIndex プロパティなどの他のプロパティが使用されます。id プロパティに値を割り当てると、テスト スクリプトを読みやすくすることができます。

オブジェクトの重複の回避

自動エージェントの処理は、実行中にオブジェクト インスタンスの一部のプロパティが変更されないことを前提としています。実行時に Silk Test Workbench によってオブジェクト名として使用されている Flex コンポーネント プロパティを変更すると、予期しない結果が発生する可能性があります。たとえば、automationName プロパティのない Button コントロールを作成し、最初は label プロパティに値を設定しないで、その後、label プロパティに値を設定した場合、問題が発生することがあります。この場合、Silk Test Workbench では、automationName プロパティが設定されていない場合は Button コントロールを識別するためにコントロールの label プロパティの値が使用されます。あとから label プロパティの値を設定したり、既存の label の値を変更すると、Silk Test Workbench ではこのオブジェクトを新しいオブジェクトとして識別し、既存のオブジェクトを参照しなくなります。

重複オブジェクトを回避するには：

- エージェントにおいて、オブジェクトの識別にどのプロパティが使用されているかを理解し、実行時にそれらのプロパティを変更しないようにします。
- 記録されたスクリプトに含まれているすべてのオブジェクトに対して、人間が理解しやすい一意の id プロパティまたは automationName プロパティを設定します。


Apache Flex アプリケーションのカスタム属性

Apache Flex アプリケーションは、あらかじめ定義されたプロパティ automationName を使用して、次のように Apache Flex コントロールに対して安定した識別子を指定します。

```
<?xml version="1.0" encoding="utf-8"?>
  <s:Group xmlns:fx="http://ns.adobe.com/mxml/2009"
    xmlns:s="library://ns.adobe.com/flex/spark"
    xmlns:mx="library://ns.adobe.com/flex/mx" width="400" height="300">
    <fx:Script>
      ...
    </fx:Script>
    <s:Button x="247" y="81" label="Button" id="button1" enabled="true"
click="button1_clickHandler(event)"
      automationName="AID_buttonRepeat"/>
    <s:Label x="128" y="123" width="315" height="18" id="label1" verticalAlign="middle"
      text="awaiting your click" textAlign="center"/>
  </s:Group>
```

Apache Flex アプリケーションのロケータは次のようになります。


```
...//SparkApplication//SparkButton[@caption='AID_buttonRepeat'
```

 **注目:** Apache Flex アプリケーションの場合、Silk Test Workbench では automationName はロケータ属性 caption に常にマップされます。automationName 属性が指定されていない場合、Silk Test Workbench は属性 ID をロケータ属性 caption にマップします。

Flex の AutomationName プロパティと AutomationIndex プロパティ

Flex オートメーション API には、automationName プロパティと automationIndex プロパティが用意されています。automationName を指定すると、Silk Test Workbench では、記録されたウィンドウ宣言の名前としてこの値が使用されます。わかりやすい名前を指定すると、そのオブジェクトを Silk Test Workbench で識別しやすくなります。ベストプラクティスとして、アプリケーションのテストに含まれているすべてのオブジェクトの automationName プロパティに値を設定することをお勧めします。

automationIndex プロパティを使用して、オブジェクトに対して一意のインデックス値を割り当てます。たとえば、2つのオブジェクトが同じ名前を共有している場合は、インデックス値を割り当てて、2つのオブジェクトを識別します。

 **注:** Silk Test Flex オートメーション SDK は、Flex のオートメーション API に基づいています。Silk Test オートメーション SDK は、Flex のオートメーション API でサポートされているものと同じコンポーネントが同様にサポートされます。たとえば、オートメーションコードを使用してアプリケーションがコンパイルされ、連続的に SWF ファイルが読み込まれる場合、メモリリークが発生して、最終的にアプリケーションでメモリが不足します。Flex Control Explorer サンプルアプリケーションは、この問題の影響を受けます。回避策として、Explorer が読み込むアプリケーションの SWF ファイルをオートメーションライブラリを使用してコンパイルしない方法があります。たとえば、Explorer のメインアプリケーションのみをオートメーションライブラリを使用してコンパイルします。SWFLoader の代わりにモジュールローダーを使用する方法もあります。Flex オートメーション API の詳細については、『Apache Flex リリースノート』を参照してください。

Flex クラス定義ファイル

クラス定義ファイルには、インストルメント化されたすべての Flex コンポーネントについての情報が含まれています。このファイルでは、記録中にイベントを送信でき、再生中にイベントを受け取ることができるコンポーネントについての情報が提供されます。クラス定義ファイルには、サポートされているプロパティの定義も含まれています。

Silk Test には、Flex の共通コントロールおよび特殊化されたコントロールのすべてのクラス、イベント、およびプロパティを記述するいくつかの XML ファイルが含まれています。これらの XML ファイルは、`<Silk_Test_install_directory>%ng%agent%plugins%com.borland.fastxd.techdomain.flex.agent_<バージョン>%config%automationEnvironment` フォルダにあります。

独自の XML ファイルを提供する場合は、XML ファイルをこのフォルダにコピーする必要があります。Silk Test のエージェントが起動して Apache Flex のサポートを初期化するとき、このディレクトリの内容が読み込まれます。

XML ファイルの基本的な構造は以下のとおりです。

```
<TypeInfoInformation>
<ClassInfo>
<Implementation />
<Events>
<Event />
...
</Events>
<Properties>
<Property />
...
</Properties>
</ClassInfo>
</TypeInfoInformation>
```

Flex の automationName プロパティの設定

automationName プロパティは、テストに表示されるコンポーネント名を定義します。このプロパティのデフォルト値は、コンポーネントの種類に応じて異なります。たとえば、Button コントロールの automationName は、Button コントロールのラベルです。automationName がコントロールの id プロパティと同じ場合もありますが、常に同じであるわけではありません。

一部のコンポーネントでは、automationName プロパティの値は、Flex によってそのコンポーネントを認識しやすい属性に設定されています。これにより、テスト担当者は、テストでコンポーネントを認識しやすくなります。通常、テスト担当者は、アプリケーションの基になるソースコードにアクセスできないため、コントロールの表示されるプロパティによってそのコントロールを認識できるようにすることは有用です。たとえば、「Process Form Now」というラベルが設定された Button は、テストで FlexButton("Process Form Now") と表示されます。

新しいコンポーネントを実装する場合や、既存のコンポーネントから派生する場合は、automationName プロパティのデフォルト値をオーバーライドできます。たとえば、UIComponent では、automationName の値は、デフォルトでコンポーネントの id プロパティに設定されます。ただし、一部のコンポーネントでは、独自の方法を使用して値が設定されます。たとえば、Flex Store サンプルアプリケーションでは、コンテナを使用して製品のサムネイルが作成されています。コンテナのデフォルトの automationName はコンテナの id プロパティと同じ値となるため、あまり役立ちません。そのため、Flex Store では、製品のサムネイルを生成するカスタムコンポーネントで明示的に automationName を製品名に設定して、アプリケーションをテストしやすくしています。

例

以下の CatalogPanel.mxml カスタム コンポーネントの例では、automationName プロパティの値をカタログに表示される項目名に設定しています。これにより、デフォルトのオートメーション名を使用するよりもサムネイルを認識しやすくなります。

```
thumbs[i].automationName = catalog[i].name;
```

例

以下の例では、ComboBox コントロールの automationName プロパティを「Credit Card List」に設定しています。このように設定すると、通常、テストツールでは、スクリプトにおいて id プロパティではなく「Credit Card List」を使用して ComboBox が識別されます。

```
<?xml version="1.0"?>
<!-- at/SimpleComboBox.mxml -->
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:Script>
    <![CDATA[
      [Bindable]
      public var cards: Array = [
        {label:"Visa", data:1},
        {label:"MasterCard", data:2},
        {label:"American Express", data:3}
      ];

      [Bindable]
      public var selectedItem:Object;
    ]>
  </mx:Script>
  <mx:Panel title="ComboBox Control Example">
    <mx:ComboBox id="cb1" dataProvider="{cards}"
      width="150"
      close="selectedItem=ComboBox(event.target).selectedItem"
      automationName="Credit Card List"
    />
    <mx:VBox width="250">
      <mx:Text width="200" color="blue" text="Select a type of credit
card." />
      <mx:Label text="You selected: {selectedItem.label}"/>
      <mx:Label text="Data: {selectedItem.data}"/>
    </mx:VBox>
  </mx:Panel>
</mx:Application>
```

automationName プロパティの値を設定すると、オブジェクト名が実行時に変更されないことが保証されます。このことは、予期しない結果の回避に役立ちます。

automationName プロパティの値を設定すると、テストでは、デフォルト値ではなく、その値が使用されます。たとえば、Silk Test Workbench では、デフォルトで、スクリプトにおいて Button コントロールの label プロパティがボタンの名前として使用されます。この場合、ラベルが変更されると、スクリプトが動作しなくなります。automationName プロパティの値を明示的に設定することによって、このような事態を回避できます。

ラベルがなく、アイコンがあるボタンは、インデックス番号によって記録されます。この場合は、automationName プロパティをわかりやすい文字列に設定して、テスト担

当者がスクリプトでボタンを認識できるようにします。automationName プロパティの値を設定したあとは、コンポーネントのライフ サイクル全体を通して値を変更しないでください。項目レンダラでは、automationName プロパティではなく automationValue プロパティを使用します。automationValue プロパティを使用するには、createAutomationIDPart() メソッドをオーバーライドして、automationName プロパティに割り当てる新しい値を返します。以下に例を示します。

```
<mx:List xmlns:mx="http://www.adobe.com/2006/mxml">
  <mx:Script>

    import mx.automation.IAutomationObject;
    override public function
    createAutomationIDPart(item:IAutomationObject):Object {
      var id:Object = super.createAutomationIDPart(item);
      id["automationName"] = id["automationIndex"];
      return id;
    }

  </mx:Script>
</mx:List>
```

このテクニックを使用して、任意のコンテナまたはリスト形式コントロールの子にインデックス値を追加します。子が自分自身のインデックスを指定する方法はありません。

名前またはインデックスを使用するように Flex の Select メソッドを設定

Flex の Select メソッドは、選択するコントロールの Name または Index を使用して記録できます。デフォルトで、Silk Test では、コントロールの名前を使用して Select メソッドが記録されます。ただし、コントロールのインデックスを使用して Select イベントを記録するように環境を変更したり、名前を使用した記録とインデックスを使用した記録を切り替えたりすることができます。

1. インデックスを使用するように変更するクラスを特定します。

以下のコントロールでは、インデックスを使用して Select イベントを記録できます。

- FlexList
- FlexTree
- FlexDataGrid
- FlexOLAPDataGrid
- FlexComboBox
- FlexAdvancedDataGrid

2. 変更するクラスに関連する XML ファイルを特定します。

上記のコントロールに関連する XML ファイルには、FlexCommonControls.xml、AdvancedDataGrid.xml、または OLAPDataGrid.xml があります。

3. 変更するクラスに関連する XML ファイルに移動します。

XML ファイルは、<Silk Test_install_directory>%ng%agent%plugins
%com.borland.fastxd.techdomain.flex.agent_<バージョン>%config%automationEnvironment フォルダにあります。

4. 対応する XML ファイルで、以下の変更を行います。

```
<ClassInfo Extends="FlexList" Name="FlexControlName"
EnableIndexedSelection="true" >
...
</ClassInfo>
```

たとえば、「FlexControlName」として「FlexList」を使用し、FlexCommonControls.xml ファイルを変更できます。

この変更では、FlexList::SelectIndex イベントの記録に IndexBasedSelection が使用されています。



注: コードの EnableIndexBasedSelection= を false に設定するか、またはこのブール値を削除すると、記録で名前が使用される設定に戻ります (FlexList::Select イベント)。

5. これらの変更内容を有効にするには、Flex アプリケーションおよび Open Agent を再起動します。

Flex コンテナのコーディング

コンテナは、ユーザー対話（ユーザーが Accordion コンテナの次のページに移動したなど）の記録、およびテスト スクリプト内でのコントロールに対する一意の場所の提供の両方の目的で使用されるため、他の種類のコントロールとは異なります。

オートメーション階層におけるコンテナの追加と削除

通常、自動テスト機能のスクリプトでは、ネストされたコンテナについての詳細情報は少なく抑えられます。テストの結果やコントロールの識別に影響がないコンテナは、スクリプトから削除されます。削除対象となるコンテナは、HBox、VBox、Canvas などの、レイアウトの目的でのみ使用されるコンテナです。ただし、ViewStack、TabNavigator、Accordion などの複数ビュー ナビゲータ コンテナで使用されている場合は削除されません。このような場合、コンテナはオートメーション階層に追加されて、ナビゲーションに使用されます。

多くの複合コンポーネントでは、Canvas や VBox などのコンテナを使用して、子が整理されます。これらのコンテナは、アプリケーション上では視覚的な効果を持ちません。この結果、これらのコンテナでは、ユーザー操作は実行されず、操作を視覚的に記録する必要もないため、通常、これらのコンテナはテストから除外されます。テストからコンテナを除外することによって、関連するテスト スクリプトが簡潔になり、読みやすくなります。

コンテナを記録から除外するには、コンテナの showInAutomationHierarchy プロパティを false に設定します（子は除外されません）。このプロパティは、UIComponent クラスによって定義されているため、UIComponent のサブクラスであるすべてのコンテナにこのプロパティが存在します。階層で表示されないコンテナの子は、階層内でそのコンテナの次に上位の親の子として表示されます。

showInAutomationHierarchy プロパティのデフォルト値は、コンテナの種類に応じて異なります。Panel、Accordion、Application、DividedBox、Form などのコンテナではデフォルト値は true であり、Canvas、HBox、VBox、FormItem などのコンテナではデフォルト値は false です。

以下の例では、VBox コンテナがテスト スクリプトの階層に組み込まれています。

```
<?xml version="1.0"?>
<!-- at/NestedButton.mxml -->
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml">
<mx:Panel title="ComboBox Control Example">
<mx:HBox id="hb">
<mx:VBox id="vb1" showInAutomationHierarchy="true">
<mx:Canvas id="c1">
<mx:Button id="b1" automationName="Nested Button 1" label="Click Me" />
</mx:Canvas>
</mx:VBox>
<mx:VBox id="vb2" showInAutomationHierarchy="true">
<mx:Canvas id="c2">
<mx:Button id="b2" automationName="Nested Button 2" label="Click Me 2" />
</mx:Canvas>
</mx:VBox>
</mx:HBox>
</mx:Panel>
</mx:Application>
```


複数ビュー コンテナ

TabNavigator や Accordion コンテナなどの複数ビュー コンテナ内の複数のタブに同じラベルを使用しないでください。同じラベルを使用することもできますが、このことは、通常、推奨 UI 設計プラクティスとして推奨されていません。このようにすると、テスト環境においてコントロールの識別に問題が発生することがあります。

Flex 自動テスト ワークフロー

Flex アプリケーションのテストの Silk Test Workbench ワークフローは、以下のとおりです。

- 自動テストの初期化
- 自動テストの記録
- 自動テストの再生

Flex 自動テストの初期化

ユーザーが Flex アプリケーションを起動すると、以下の初期化イベントが発生します。

1. オートメーション初期化コードによって、コンポーネントの委譲クラスがコンポーネントのクラスに関連付けられます。
2. コンポーネントの委譲クラスは、IAutomationObject インターフェイスを実装します。
3. AutomationManager のインスタンスがミックスインの init() メソッドで作成されます。(AutomationManager はミックスインです。)
4. SystemManager によってアプリケーションが初期化されます。コンポーネント インスタンスおよび対応する委譲インスタンスが作成されます。委譲インスタンスによって、目的のイベントに対するイベント リスナーが追加されます。
5. Silk Test Workbench FlexTechDomain はミックスインです。FlexTechDomain の init() メソッドで、FlexTechDomain が SystemManager.APPLICATION_COMPLETE イベントに登録されます。イベントを受信すると、FlexTechDomain インスタンスが作成されます。
6. FlexTechDomain インスタンスが、同じマシン上の記録および再生機能に登録する Silk Test Agent に TCP/IP ソケット経由で接続します。
7. FlexTechDomain は、自動環境についての情報を要求します。この情報は XML ファイルに格納され、Silk Test Agent から FlexTechDomain に転送されます。

Flex 自動テストの記録

ユーザーが Silk Test Workbench で Flex アプリケーションの新しいテストを記録すると、以下のイベントが発生します。

1. Silk Test Workbench によって Silk Test Agent が呼び出されて、記録が開始されます。Agent は、このコマンドを FlexTechDomain インスタンスに転送します。
2. FlexTechDomain は、beginRecording() を呼び出すことによって、AutomationManager に対して記録の開始を通知します。AutomationManager は、SystemManager からの AutomationRecordEvent.RECORD イベントに対するリスナーを追加します。
3. ユーザーがアプリケーションを操作します。たとえば、ユーザーが Button コントロールをクリックしたとします。
4. ButtonDelegate.clickEventHandler() メソッドによって、プロパティとしてクリック イベントと Button のインスタンスが指定された AutomationRecordEvent イベントがディスパッチされます。
5. AutomationManager は、XML 環境情報に基づいて、クリック イベントのどのプロパティを格納するかを決定します。値が適切な型または書式に変換されます。記録イベントがディスパッチされます。
6. FlexTechDomain イベントハンドラがイベントを受信します。AutomationManager.createID() メソッドが呼び出されて、ボタンの AutomationID オブジェクトが作成されます。このオブジェクトは、オブジェクト識別用の構造体を提供します。AutomationID 構造体は、AutomationIDParts の配列になっています。AutomationIDParts は、IAutomationObject を使用して作成されます。(Button コントロールの UIComponent.id、automationName、automationValue、childIndex、および label プロ

パーティが読み込まれて、オブジェクトに格納されます。XML 情報に、label プロパティを Button の識別に使用できることが指定されているため、label プロパティが使用されます。)

7. FlexTechDomain は、AutomationManager.getParent() メソッドを使用して、Button の論理的な親を取得します。アプリケーション レベルまでの各レベルで、親コントロールの AutomationIDParts オブジェクトが収集されます。
8. すべての AutomationIDParts が AutomationID オブジェクトの一部として組み込まれます。
9. FlexTechDomain は、Silk Test Workbench への呼び出しでこの情報を送信します。
10. ユーザーが記録を停止すると、FlexTechDomain.endRecording() メソッドが呼び出されます。

Flex 自動テストの再生

ユーザーが Silk Test Workbench で **再生** ボタンをクリックすると、以下のイベントが発生します。

1. 各スクリプト呼び出しにおいて、Silk Test Workbench は Silk Test Agent に接続し、実行されるスクリプト呼び出しの情報を送信します。この情報には、完全なウィンドウ宣言、イベント名、およびパラメータが含まれています。
2. Silk Test Agent は、その情報を FlexTechDomain に転送します。
3. FlexTechDomain は、ウィンドウ宣言情報と共に AutomationManager.resolveIDToSingleObject を使用します。AutomationManager は、説明情報 (automationName、automationIndex、id など) に基づいて、解決したオブジェクトを返します。
4. Flex コントロールが解決されると、FlexTechDomain は AutomationManager.replayAutomatableEvent() を呼び出して、イベントを再生します。
5. AutomationManager.replayAutomatableEvent() メソッドによって、委譲クラスの IAutomationObject.replayAutomatableEvent() メソッドが呼び出されます。委譲では、IAutomationObjectHelper.replayMouseEvent() メソッド (または replayKeyboardEvent() などの他のいずれかの再生メソッド) を使用してイベントが再生されます。
6. スクリプトに検証がある場合、FlexTechDomain は AutomationManager.getProperties() を呼び出して、検証する必要がある値にアクセスします。

Apache Flex アプリケーションのスタイル

Apache Flex 3.x で開発されたアプリケーションについて、Silk Test Workbench ではスタイルとプロパティを区別しません。この結果、スタイルはプロパティとして公開されます。ただし、Apache Flex 4.x の Spark という接頭辞が付いているすべての新しい Flex コントロール (SparkButton など) では、スタイルがプロパティとして公開されません。この結果、Flex 4.x コントロールの GetProperty() メソッドおよび GetPropertyList() メソッドでは color や fontSize などのスタイルが返されず、text や name などのプロパティのみが返されます。

GetStyle(string styleName) メソッドは、スタイルの値を文字列として返します。どのようなスタイルが存在するかを確認するには、次の Adobe ヘルプを参照してください: http://help.adobe.com/ja_JP/FlashPlatform/reference/actionsript/3/package-detail.html。

スタイルが設定されていない場合は、再生中に StyleNotSetException が発生します。

FlexTree などの Flex 3.x コントロールでは、GetProperty() を使用してスタイルを取得できます。GetStyle() を使用することもできます。Flex 3.x コントロールでは、GetProperty() メソッドと GetStyle() メソッドの両方が動作します。

色スタイルの計算

Flex では、色は数値として表されます。色は、以下の式を使用して計算できます。

```
red*65536 + green*256 + blue
```

例

以下のスクリプト例では、Spark アプリケーションの ButtonBar がフォント サイズ 12 を使用しているかどうかを検証しています。

```
Imports SilkTest.Ntf.Flex

Public Module Main
    Dim _desktop As Desktop = Agent.Desktop

    Public Sub Main()
        Dim Application As SparkApplication
        Dim ButtonBar As SparkButtonBar
        Application = _desktop.Find( "/BrowserApplication//
        BrowserWindow//
        SparkApplication" )
        ButtonBar = Application.SparkButtonBar()

        Workbench.Verify(ButtonBar.GetStyle( "fontSize" ), "12" )
    End Sub
End Module
```

Adobe Flash Player のセキュリティ制約に対応するための Flex アプリケーションの構成

Adobe Flash Player 10 では、セキュリティ モデルが以前のバージョンから変更されています。Flash Player を使用するテストを記録する場合、記録は想定どおりに動作します。ただし、テストを再生する場合は、特定の状況で高レベルのクリックが行われると、予期しない結果が発生します。たとえば、**ファイル参照** ダイアログ ボックスをプログラムから開くことができません。このシナリオの再生を試みると、セキュリティ制約が原因でテストに失敗します。

このセキュリティ制約を回避するには、ダイアログ ボックスを開くボタンに対して低レベルのクリックを実行します。低レベルのクリックを作成するには、Click メソッドにパラメータを追加します。

たとえば、SparkButton::Click() の代わりに SparkButton::Click(MouseButton.Left) を使用します。パラメータを指定しない Click() は高レベルのクリックとして再生され、パラメータを指定したクリック (ボタンなど) は低レベルのクリックとして再生されます。

1. Flash Player を使用するテストを記録します。
2. Click メソッドに移動して、パラメータを追加します。
たとえば、**ファイルを開く** ダイアログ ボックスを開くには、以下のように指定します：

```
SparkButton("@caption='Open File Dialog...']").Click(MouseButton.Left)
```

○
テストを再生すると、想定どおりに動作します。

Apache Flex アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Flex アプリケーションがサポートする属性は次のとおりです。

- automationName
- caption (automationName と同様)

- automationClassName (FlexButton など)
- className (実装クラスの完全修飾名。 mx.controls.Button など)
- automationIndex (FlexAutomation のビューでのコントロールのインデックス。 index:1 など)
- index (automationIndex と同様。ただし、接頭辞はなし。 1 など)
- id (コントロールの ID)
- windowId (id と同様)
- label (コントロールのラベル)
- すべての動的ロケータ属性



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

Silk Test Workbench が Apache Flex コントロールを認識できない理由

Web サーバーを通してアクセスしている Apache Flex アプリケーションのコントロールを Silk Test Workbench が認識できない場合、以下のことを行ってみてください。

- Adobe オートメーション ライブラリと Apache Flex バージョン用に適した FlexTechDomain.swc を使用して、Apache Flex アプリケーションをコンパイルします。
- 実行時ローディングを使用します。
- Apache Flex アプリケーションを空の id 属性を使用して埋め込んでいると、Apache Flex コントロールは認識されません。

Java AWT/Swing のサポート

Silk Test Workbench は、Java AWT/Swing コントロールを使用するアプリケーションまたはアプレットのテストを組み込みでサポートしています。Java AWT/Swing を使用するアプリケーションまたはアプレットを設定すると、Silk Test Workbench は標準の AWT/Swing コントロールのテストのサポートを組み込みで提供します。



注: Java AWT/Swing アプリケーションまたはアプレットに埋め込まれた Java SWT コントロールや、Java SWT アプリケーションに埋め込まれた Java AWT/Swing コントロールもテストできます。



注: イメージ クリックの記録は、Java AWT/Swing コントロールを使用するアプリケーションまたはアプレットではサポートされません。

サンプル アプリケーション

Silk Test は、サンプルの Swing テスト アプリケーションを提供しています。サンプル アプリケーションを <http://supportline.microfocus.com/websync/SilkTest.aspx> からダウンロードしてインストールします。サンプル アプリケーションをインストールしたあと、(Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > Sample Applications > Java Swing > Swing Test Application**、または (Microsoft Windows 10) **スタート > Silk** をクリックします。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリース ノート](#)』を参照してください。

サポートするコントロール

Java AWT/Swing のテストで使用可能なコントロールの完全な一覧については、「[Java Swing クラス リファレンス](#)」を参照してください。

Java AWT/Swing アプリケーションの属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。

Java AWT/Swing でサポートされる属性には以下のものがあります。

- caption
- priorlabel : 隣接するラベル フィールドのテキストによってテキスト入力フィールドを識別します。通常、フォームのすべての入力フィールドに、入力の目的を説明するラベルがあります。caption のないコントロールの場合、自動的に属性 **priorlabel** がロケータに使用されます。コントロールの **priorlabel** 値 (テキスト入力フィールドなど) には、コントロールの左側または上にある最も近いラベルの caption が使用されます。
- name
- accessibleName
- *Swing* のみ : すべてのカスタム オブジェクトの定義属性は、ウィジェットに `putClientProperty("propertyName", "propertyValue")` で設定されます。



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

Java メソッドの動的な呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。



注: 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。

オブジェクトの動的メソッドは `Invoke` メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、`GetDynamicMethodList` メソッドを使用します。

オブジェクトの複数の動的メソッドは `InvokeMethods` メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、`GetDynamicMethodList` メソッドを使用します。

動的プロパティの取得には `GetProperty` メソッドを、動的プロパティの設定には `SetProperty` メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、`GetPropertyList` メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを `String` 型の入力パラメータとして設定する必要がある `SetTitle` というメソッドを呼び出すには、次のように入力します :

```
control.Invoke("SetTitle", "my new title")
```



注: 通常、ほとんどのプロパティは読み取り専用で、設定できません。



注: ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、`Reflection` を使用します。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- SWT、AWT、または Swing ウィジェットのすべてのパブリック メソッド
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- プリミティブ型 (boolean、integer、long、double、string)
プリミティブ型 (int など) とオブジェクト タイプ (java.lang.Integer など) の両方がサポートされます。プリミティブ型は必要に応じて拡大変換されます。たとえば、long が必要な場所で int を渡すことができます。
- 列挙型
列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが列挙型 java.sql.ClientInfoStatus のパラメータを必要とする場合、次の文字列値を使用できます： REASON_UNKNOWN、REASON_UNKNOWN_PROPERTY、REASON_VALUE_INVALID、REASON_VALUE_TRUNCATED
- リスト
リスト、配列、または可変長引数のパラメータを持つメソッドを呼び出すことができます。リストの要素がターゲットの配列型に代入可能の場合、配列型への変換は自動的に行われます。
- その他のコントロール
コントロール パラメーターは、TestObject として渡したり、返したりできます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

Java AWT/Swing テクノロジ ドメインでの priorLabel の判別

Java AWT/Swing テクノロジ ドメインで priorLabel を判別するには、ターゲット コントロールと同じウィンドウ内のすべてのラベルおよびグループを考慮する必要があります。判別の基準は、次のとおりです。

- priorLabel の候補とみなされるのは、コントロールの上または左にあるラベル、およびコントロールが属しているグループのみです。
- コントロールの親が JViewport または ScrollPane の場合、アルゴリズムはこのコントロールを含むウィンドウが親であるかのように機能し、外側の要素はどれも関連しないとみなされます。
- 最も単純なケースでは、コントロールに最も近いラベルが priorLabel として使用されます。
- 2 つのラベルがコントロールから等距離にあり、1 つがコントロールの左、もう 1 つが上にある場合は、左側のラベルが優先します。
- 適したラベルがない場合は、最も近いグループのキャプションが使用されます。

Oracle Forms のサポート

この機能がサポートされるのは、Open Agent を使用している場合のみです。

Silk Test Workbench は、Oracle Forms ベースのアプリケーションのテストを組み込みでサポートしません。

 **注:** コントロールによっては、Silk Test Workbench は低レベルの記録のみをサポートします。

サポート対象バージョン、既知の問題、回避策の詳細については、「[リリースノート](#)」を参照してください。

Oracle Forms テストの前提条件

Oracle Forms を使ってビルドしたアプリケーションをテストするには、次の前提条件を満たす必要があります。


- 次世代 Java プラグインを有効化する必要があります。この設定は、デフォルトで有効になっています。**Java コントロールパネル** で設定を変更できます。次世代 Java プラグインの詳細については、Java のドキュメントを参照してください。
- Java セキュリティ ダイアログをテストの実行中に表示されないようにするには、アプレットにサインする必要があります。
- Micro Focus は、Oracle Forms 開発者が Names プロパティを有効にすることを推奨します。このプロパティを有効にすると、Oracle Forms ランタイムは内部の名前を公開します。この名前は、コントロールの開発者が、コントロールの Name プロパティとしてコントロールに対して指定したものです。そうでない場合、Silk Test Workbench は、コントロールのクラス名とインデックスから構成された値が、Silk Test Workbench Name 属性として設定されます。これを使用すると、Silk Test Workbench はコントロールに対して安定したロケータを生成できます。

Oracle Forms アプリケーションの属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジードメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。

Oracle Forms がサポートする属性は次のとおりです。

- **priorlabel** : 隣接するラベルフィールドのテキストによってテキスト入力フィールドを識別します。通常、フォームのすべての入力フィールドに、入力の目的を説明するラベルがあります。caption のないコントロールの場合、自動的に属性 **priorlabel** がロケータに使用されます。コントロールの **priorlabel** 値 (テキスト入力フィールドなど) には、コントロールの左側または上にある最も近いラベルの caption が使用されます。
- name
- accessibleName

 **注:** 属性名は、大文字小文字が区別されます (モバイルアプリケーションを除く。モバイルアプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

Java SWT と Eclipse RCP のサポート

Silk Test は、SWT (Standard Widget Toolkit) コントロールのウィジェットを使用したアプリケーションのテストの組み込みサポートを提供します。Java SWT/RCP アプリケーションを設定すると、Silk Test は標準の Java SWT/RCP コントロールのテストのサポートを組み込みで提供します。

Silk Test では、以下をサポートしています。

- Java AWT/Swing アプリケーションに埋め込まれた Java SWT コントロール、および Java SWT アプリケーションに埋め込まれた Java AWT/Swing コントロールのテスト。
- Java SWT アプリケーションのテスト。

- レンダリングに SWT ウィジェットを使用した Eclipse ベースのアプリケーション。 Silk Test は、Eclipse IDE ベース、および RCP ベースの両方のアプリケーションをサポートします。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

サポートするコントロール

SWT テストで使用できるウィジェットの完全なリストについては、「[Java SWT クラス リファレンス](#)」を参照してください。

Java SWT クラス リファレンス

Java SWT アプリケーションを設定すると、Silk Test Workbench は標準の Java SWT コントロールのテストのサポートを組み込みで提供します。

Java SWT カスタム属性

カスタム属性をテスト アプリケーションに追加して、テストをより安定させることができます。たとえば、Java SWT では、GUI を実装する開発者が属性 ('silkTestAutomationId' など) をウィジェットに対して定義することによって、アプリケーション内でそのウィジェットを一意に識別することができます。これにより、Silk Test Workbench を使用するテスト担当者は、その属性 (この場合は 'silkTestAutomationId') をカスタム属性のリストに追加すると、その一意の ID によってコントロールを識別できるようになります。カスタム属性を使用すると、caption や index のような他の属性よりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は定義済みのウィジェットより前に他のウィジェットが追加されると変更されるためです。

複数のオブジェクトに同じカスタム属性の値が割り当てられた場合は、そのカスタム属性を呼び出したときにその値を持つすべてのオブジェクトが返されます。たとえば、一意の ID として 'loginName' を 2 つの異なるテキスト フィールドに割り当てた場合は、'loginName' 属性を呼び出したときに、両方のフィールドが返されます。

Java SWT の例

以下のコードを使用して、テストするアプリケーションにボタンを作成する場合：

```
Button myButton = Button(parent, SWT.NONE);  
  
myButton.setData("SilkTestAutomationId", "myButtonId");
```

テストの XPath クエリ文字列に属性を追加するには、以下のクエリを使用します。

```
Dim button =  
desktop.PushButton("@SilkTestAutomationId='myButton'")
```


Java SWT アプリケーションをカスタム属性のテストに対して有効化にするには、開発者はカスタム属性をアプリケーションに含める必要があります。属性を含めるには `org.swt.widgets.Widget.setData(String key, Object value)` メソッドを使用します。

Java SWT アプリケーションの属性

ローケターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジ ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ローケターがテスト内のオブジェクトを識別する方法が決定されます。


Java SWT がサポートする属性は次のとおりです。

- caption
- すべてのカスタム オブジェクト定義属性

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ローケター属性は、ワイルドカード ? および * をサポートしています。

Java メソッドの動的な呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある SetTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- SWT、AWT、または Swing ウィジェットのすべてのパブリック メソッド
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- プリミティブ型 (boolean、integer、long、double、string)

プリミティブ型 (int など) とオブジェクト タイプ (java.lang.Integer など) の両方がサポートされます。プリミティブ型は必要に応じて拡大変換されます。たとえば、long が必要な場所で int を渡すことができます。

- 列挙型

列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが列挙型 java.sql.ClientInfoStatus のパラメータを必要とする場合、次の文字列値を使用できます： REASON_UNKNOWN、REASON_UNKNOWN_PROPERTY、REASON_VALUE_INVALID、REASON_VALUE_TRUNCATED

- リスト
リスト、配列、または可変長引数のパラメータを持つメソッドを呼び出すことができます。リストの要素がターゲットの配列型に代入可能な場合、配列型への変換は自動的に行われます。
- その他のコントロール
コントロールパラメーターは、TestObject として渡したり、返したりできます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

Java SWT と Eclipse アプリケーションのトラブルシューティング

一部の SWTTree メソッドが低レベル再生時に再生されない

低レベル再生を使用したとき、Expand や Collapse などの一部の SWTTree メソッドが再生されません。

この問題を解決するには、再生モードを **デフォルト** に設定します。詳細については、「全般再生オプションの設定」を参照してください。

SWTTree の非表示ノードの選択

低レベル再生を使用したとき、Silk Test Workbench は SWTTree の非表示ノードを操作できません。

この問題を解決するには、再生モードを **デフォルト** に設定します。詳細については、「全般再生オプションの設定」を参照してください。

モバイル アプリケーションのテスト

Silk Test Workbench では、ネイティブ モバイル アプリケーション（アプリ）およびモバイル Web アプリケーションを自動的にテストすることができます。Silk Test Workbench を使用してモバイル アプリケーションを自動的にテストすることには、次のメリットがあります。

- モバイル アプリケーションのテスト時間を大幅に減少させることができます。
- テストを一旦作成すれば、数多くの異なるデバイスやプラットフォーム上でモバイル アプリケーションをテストできます。
- エンタープライズ モバイル アプリケーションに要求される信頼性とパフォーマンスを確保できます。
- QA チームのメンバーおよびモバイル アプリケーションの開発者の効率を向上できます。
- モバイル アプリケーションは、多くのモバイル デバイスとプラットフォームで動作することを要求されるため、アジャイルにフォーカスした開発環境にとって手動テストは十分効率的とは言えない場合があります。



注: Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「ライセンス情報」を参照してください。



注: Silk Test Workbench は、Android および iOS デバイスの両方でのモバイル アプリのテストをサポートします。

モバイル アプリケーションのテストをサポートするオペレーティング システムとサポートするブラウザについての情報は、『[リリース ノート](#)』を参照してください。

Android

Silk Test Workbench では、Android デバイスまたは Android エミュレータ上のモバイル アプリケーションをテストすることができます。

Android 上のモバイル アプリケーションのテストにおける前提条件

Android デバイスや Android エミュレータ上のモバイル アプリケーションをテストする前に、次の前提条件を満たしていることを確認してください。


- ネイティブ モバイル アプリに Web ビューを追加したハイブリッド アプリを作成した場合は、アプリを Silk Test Workbench でテスト可能にするために、次のコードをアプリに追加してください。

```
WebView.setWebContentsDebuggingEnabled(true);  
webView.getSettings().setJavaScriptEnabled(true);
```

- Silk Test Workbench では、Android 4.4 のデバイス画面のライブ ビュー表示をサポートしていません。このため、Android 5 以降を使用することを Micro Focus は推奨しています。

Android 上のモバイル アプリケーションのテスト

物理 Android デバイスや Android エミュレータ上のモバイル アプリケーションをテストするには、次のタスクを実行します。

1. Android 上のモバイル アプリケーションのテストにおける前提条件を満たすことを確認します。
詳細については、「[Android 上のモバイル アプリケーションのテストにおける前提条件](#)」を参照してください。
2. Android エミュレータ上のモバイル アプリケーションをテストするには、Silk Test Workbench 用にエミュレータを設定します。
詳細については、「[Silk Test 用に Android エミュレータを設定する](#)」を参照してください。
3. Silk Test Workbench をインストールしたマシンで Android エミュレータを開始するか、デバイスを接続します。
4. 物理 Android デバイスを初めて使用するマシンでテストする場合、適切な Android USB ドライバをマシンにインストールします。
詳細については、「[USB ドライバのインストール](#)」を参照してください。
5. 物理 Android デバイス上でモバイル アプリケーションをテストする場合は、Android デバイスの USB デバッグを有効化します。
詳細については、「[USB デバッグの有効化](#)」を参照してください。
6. モバイル アプリケーション用の Silk Test Workbench プロジェクトを作成します。
7. モバイル アプリケーション用のテストを作成します。
8. テストで実行する操作を記録します。記録 ウィンドウを開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。
9. モバイル Web アプリケーションをテストするには：
 - a) **Web** タブを選択します。
 - b) 使用するモバイル ブラウザーを選択します。
 - c) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。
10. ネイティブ モバイル アプリケーションまたはハイブリッド アプリケーションをテストするには：
 **注:** Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「[ライセンス情報](#)」を参照してください。
 - a) **モバイル** タブを選択します。
 - b) アプリをテストするモバイル デバイスをリストから選択します。

c) ネイティブ モバイル アプリケーションを選択します。

- モバイル デバイスまたはエミュレータ上にアプリをインストールする場合は、**参照** をクリックしてアプリ ファイルを選択するか、**アプリ ファイル** テキスト フィールドにアプリ ファイルへの完全パスを入力します。このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
- Android デバイス上に既にインストールされているアプリを使用する場合は、**パッケージ/アクティビティ** リストからアプリを選択するか、**パッケージ/アクティビティ** フィールドにパッケージとアクティビティを指定します。
- UFT Mobile 上で利用可能なアプリを使用する場合は、**アプリ識別子** を指定します。

11OK をクリックします。

Android デバイスまたはエミュレータの画面が、テスト中にロックされないようにしてください。マシンに接続中にデバイスがロックされないようにするには、**開発者向けオプション** を開きます。**スリープモードにしない** または **充電中に画面をスリープにしない** をチェックします。

12記録 ウィンドウを使用して、モバイル アプリケーションに対するテストを記録します。

詳細については、「[モバイル アプリケーションの記録](#)」を参照してください。

13すべての操作の記録を終えたら、記録を停止します。

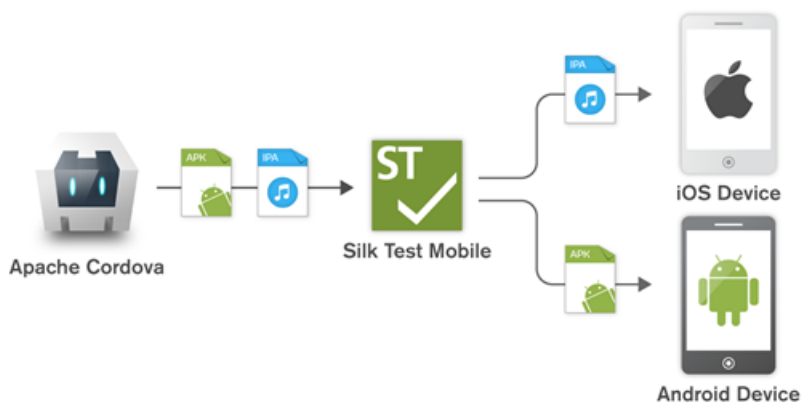
14テストを再生します。

15テスト結果を分析します。

Android 上のハイブリッド アプリケーションのテスト

ハイブリッド アプリケーション (アプリ) は、デバイス上で実行されるネイティブ アプリケーションのようなアプリですが、HTML5、CSS、JavaScript などの Web テクノロジーを使用して記述されたアプリです。

Silk Test Workbench は、ネイティブ コンテナに埋め込まれた単一の Web ビューで構成されたデバッグハイブリッド アプリのテストにする完全なブラウザ サポートを提供します。このようなハイブリッドアプリの一般的な例は、Apache Cordova アプリケーションです。



非デバッグハイブリッド アプリをテスト可能にするには、次のコードをアプリに追加してアプリをリモートデバッグできるようにします。

```
WebView.setWebContentsDebuggingEnabled(true);  
webView.getSettings().setJavaScriptEnabled(true);
```

リモートデバッグを有効化しない非デバッグハイブリッドアプリや、複数の Web ビューを含んだハイブリッドアプリをテストするには、Silk Test Workbench フォールバックサポートを有効化するために、オプション `OPT_ENABLE_MOBILE_WEBVIEW_FALLBACK_SUPPORT` を `TRUE` に設定します。詳細については、「[詳細オプションの設定](#)」を参照してください。フォールバックサポートを有効化すると、Silk Test Workbench は Web ビューのコントロールをブラウザ コントロールではなく、ネイティブ モバイ


ルコントロールとして解決して処理します。たとえば、以下のコードは、ブラウザー サポートを使用したときのリンクのクリックです。

```
' VB .NET code
Agent.SetOption(Options.EnableMobileWebviewFallbackSupport, False)
_desktop.DomLink("//BrowserApplication//BrowserWindow//INPUT[@id='email']").Click()
```

フォールバック サポートを有効化すると、同じリンクをクリックするコードは次のようになります。

```
' VB .NET code
Agent.SetOption(Options.EnableMobileWebviewFallbackSupport, True)
_desktop.Find("//BrowserApplication//BrowserWindow//MobileTextField[@resource-id='email']").Click()
```

Silk Test Workbench は、Chrome リモート デバッグをサポートする Web ビューを検出できます。Silk Test Workbench は、*com.android.webview* パッケージまたは *com.google.android.webview* パッケージ (多くの Android デバイス上のデフォルト パッケージ) のいずれかを使用した Web ビューを検出できます。

 **注:** Silk Test Workbench は、Android 4.4 以降でのハイブリッド アプリのテストをサポートします。Android でハイブリッド アプリをテストするには、Android システムの WebView バージョン 51 以降が必要です。

Android 上のハイブリッド アプリをテストする手順は、モバイル ネイティブ アプリケーションをテストする手順と同じです。詳細については、「[Android 上のモバイル アプリケーションのテスト](#)」を参照してください。

USB ドライバのインストール

モバイル アプリケーションをテストするために、ローカル マシンに最初に Android デバイスに接続するには、適切な USB ドライバをインストールする必要があります。

デバイスの製造元は、そのデバイスに必要なすべてのドライバをもった実行可能ファイルを提供している可能性があります。この場合、ローカル マシンにその実行可能ファイルをインストールするだけです。製造元がこのような実行可能ファイルを提供していない場合、マシン上にデバイスに対する単一の USB ドライバをインストールできます。

Android USB ドライバをインストールするには :

1. デバイス用の適切なドライバをダウンロードします。
たとえば、Google Nexus デバイス用の USB ドライバを検索し、インストールする場合は、<http://developer.android.com/tools/extras/oem-usb.html> を参照します。
2. Android デバイスをローカル マシンの USB ポートに接続します。
3. デスクトップ、または **Windows Explorer** から、**コンピュータ** を右クリックし、**管理** を選択します。
4. 左側のペインで、**デバイス マネージャ** を選択します。
5. 右側のペインで、**その他のデバイス** を探して展開します。
6. デバイス名 (*Nexus 5x* など) を右クリックして、**ドライバ ソフトウェアの更新** を選択します。**ハードウェアの更新ウィザード** が開きます。
7. **コンピュータを参照してドライバ ソフトウェアを検索します** を選択して、**次へ** をクリックします。
8. **参照** をクリックし、USB ドライバをダウンロードしたフォルダに移動します。
9. USB ドライバを選択します。
- 10 **次へ** をクリックしてドライバをインストールします。

USB デバッグの有効化

Android Debug Bridge (adb) 上で Android デバイスと通信するために USB デバッグを有効化します。

1. Android デバイスで設定を開きます。

2. 開発者向けオプション (Dev Settings) をタップします。

開発者向けオプションは、デフォルトでは表示されません。開発者向けオプションがデバイスの設定メニューに含まれていない場合：

- a) 画面を下にスクロールさせて、デバイスが携帯電話の場合は **端末情報** を、タブレットの場合は **タブレット情報** をタップします。
- b) 再度画面を下にスクロールさせて **ビルド番号** を 7 回タップします。

3. 開発者向けオプション ウィンドウで、**USB デバッグ** をオンにします。

4. デバイスの USB モードをデフォルトの設定である **メディア デバイス (MTP)** に設定します。


詳細については、デバイスのドキュメントを参照してください。

Android デバイスの推奨設定

Silk Test Workbench を使用したテストを最適化するために、テストしたい Android デバイスで次の設定を行ってください。

- USB デバッグを Android デバイスで有効化します。詳細については、「[USB デバッグの有効化](#)」を参照してください。
- Android デバイスは、Open Agent を実行しているマシンに、メディア デバイスとして接続されている必要があります。Android デバイスの USB モードは、**メディア デバイス (MTP)** を設定します。
- Android デバイスまたはエミュレータの画面が、テスト中にロックされないようにしてください。マシンに接続中にデバイスがロックされないようにするには、**開発者向けオプション** を開きます。**スリープモードにしない** または **充電中に画面をスリープにしない** をチェックします。
- Android の設定で、**3 ボタン ナビゲーション** を有効にしてください。**2 ボタン ナビゲーション** と **ジェスチャー ナビゲーション** はサポートしていません。

Silk Test Workbench 用に Android エミュレータを設定する

 **注:** Android エミュレータを使用する場合、Silk Test Workbench が使用している adb サーバー以外の adb サーバーが実行されてます。バージョンの異なる adb サーバーが実行中の場合、Open Agent とデバイスとの接続が不安定になったり、接続できない場合があります。このようなバージョンの不一致によるエラーを避けるには、環境変数 `SILK_ANDROID_HOME` に Android SDK ディレクトリへのパス (`C:\¥Users¥<ユーザー>¥AppData¥Local¥Android¥android-sdk` など) を指定してください。Information Service が実行中の場合、Windows のサービス マネージャーを使用して、Silk Test Information Service を再起動して更新した環境変数を適用する必要があります。環境変数が設定されていない場合は、Silk Test Workbench は Silk Test Workbench に同梱されたバージョンの adb を使用します。

Silk Test Workbench を使用して Android エミュレータ上でモバイル アプリケーションをテストする場合、テスト用にエミュレータを設定する必要があります。

1. Android SDK の最新版をインストールします。

Android SDK のインストールと設定についての詳細は、「[Get the Android SDK](#)」を参照してください。

2. Android Studio 2 以降をインストールします。



ヒント: Android Studio のインストールをスキップして、Android SDK で提供されるエミュレータを使用することもできます。ただし、エミュレータのパフォーマンスを向上させるため、Android Studio をインストールすることを、Micro Focus では推奨しています。このトピックの以下の手順では、Android Studio がインストールされていることを前提としています。

3. Android Studio から **AVD Manager** を開きます。

4. **Create Virtual Device** をクリックします。

5. 仮想デバイスを選択します。

6. **Next** をクリックします。


7. Google API を含む Android のシステム イメージをダウンロードして選択します。

8. **Next** をクリックします。

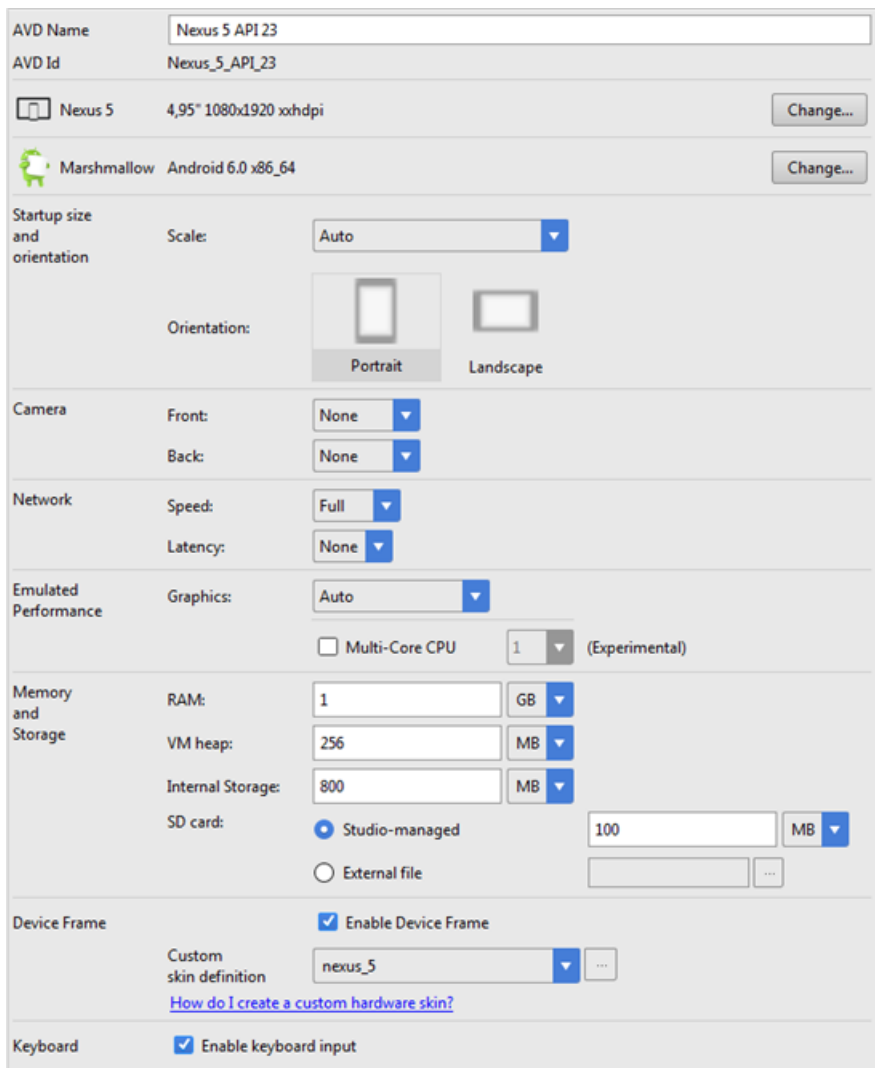
9. 要件に従って仮想デバイスを設定します。

10 **Show Advanced Settings** をクリックします。

11 エミュレータが使用する RAM サイズとヒープ領域を対象のマシンで管理可能な量に調整します。

 **ヒント:** 最低 1 GB の RAM と 256 MB のヒープ領域を使用することを、Micro Focus では推奨しています。

12 **Emulated Performance** 領域のリストで **Auto** を選択します。



The screenshot shows the AVD configuration window for a Nexus 5 device with API level 23. The 'Emulated Performance' section is expanded, showing the following settings:

- Graphics: Auto
- Multi-Core CPU: 1 (Experimental)

Other visible settings include:

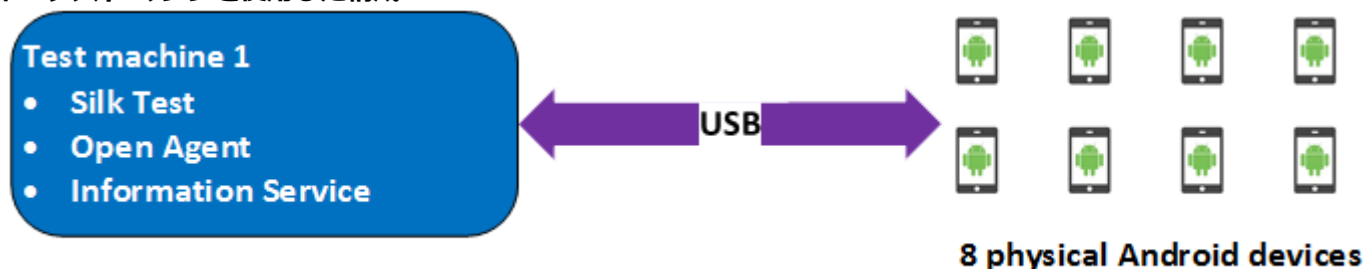
- Startup size and orientation: Scale: Auto, Orientation: Portrait
- Camera: Front: None, Back: None
- Network: Speed: Full, Latency: None
- Memory and Storage: RAM: 1 GB, VM heap: 256 MB, Internal Storage: 800 MB, SD card: Studio-managed (100 MB)
- Device Frame: Enable Device Frame (checked), Custom skin definition: nexus_5
- Keyboard: Enable keyboard input (checked)

13 **Finish** をクリックします。

並列テスト実行のテスト済みの構成

Silk Test Workbench を使用して、複数の Android デバイス上で並列に自動テストを実行することができます。並列に実行できる Android デバイスの数は利用可能なハードウェアに依存します。Micro Focus では、次のハードウェア構成でテストの実行を確認しています。

単一テストマシンを使用した構成

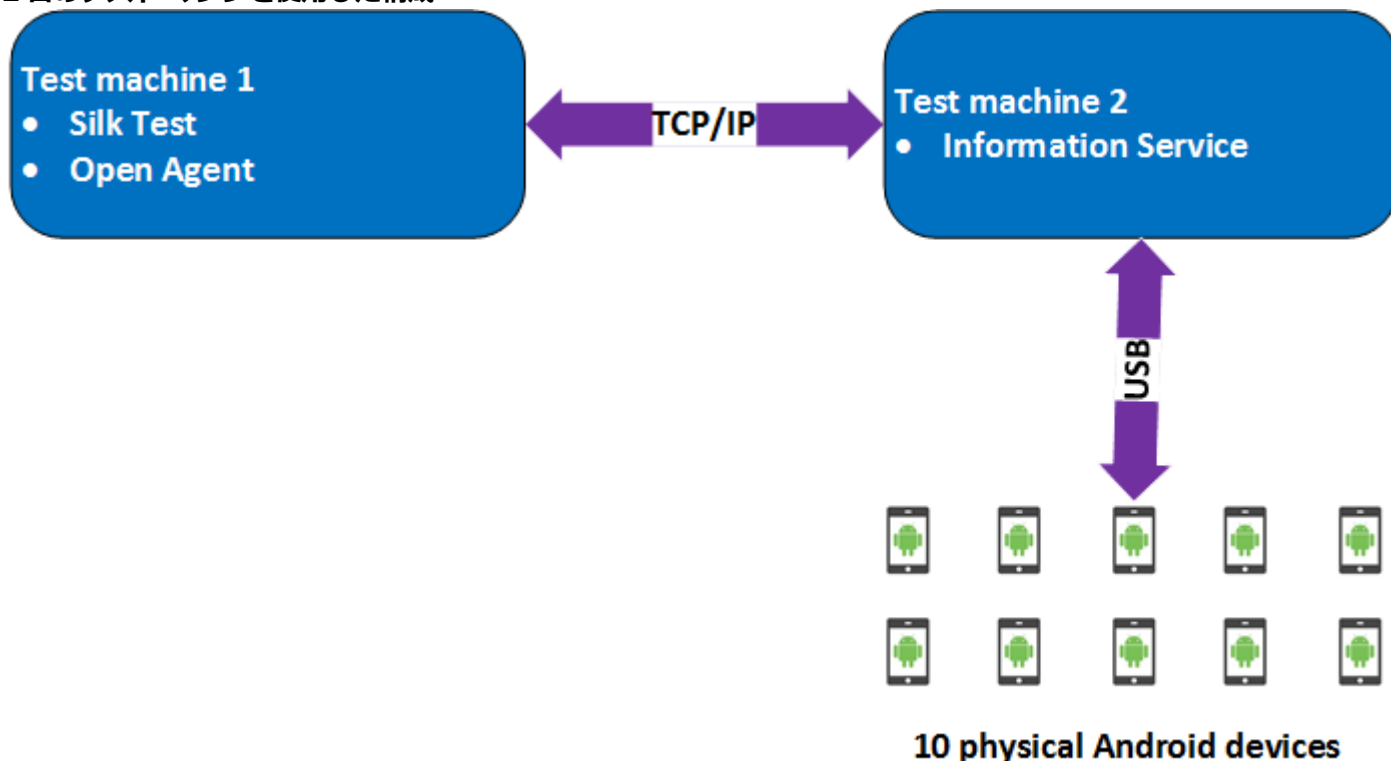


USBによりAndroidデバイスを直接接続した単一のテストマシンを使用して、最大8台の物理Androidデバイスまでの並列テストを行いました。

テストマシンは、次のハードウェア仕様のLenovo ThinkPad T450です。

- Intel® Core™ i7 - 5600U CPU @ 2.60 GHz
- 2コア(4スレッド)
- 8 GB RAM

2台のテストマシンを使用した構成



また、2台のテストマシンを使用してテストを行いました。1台にSilk Test Workbenchをインストールし、もう1台にはSilk Test Information Serviceをインストールして最初のマシンのリモートロケーションとして設定しました。このような構成を使用して、最大10台の物理Androidデバイスまでの並列テストを行いました。

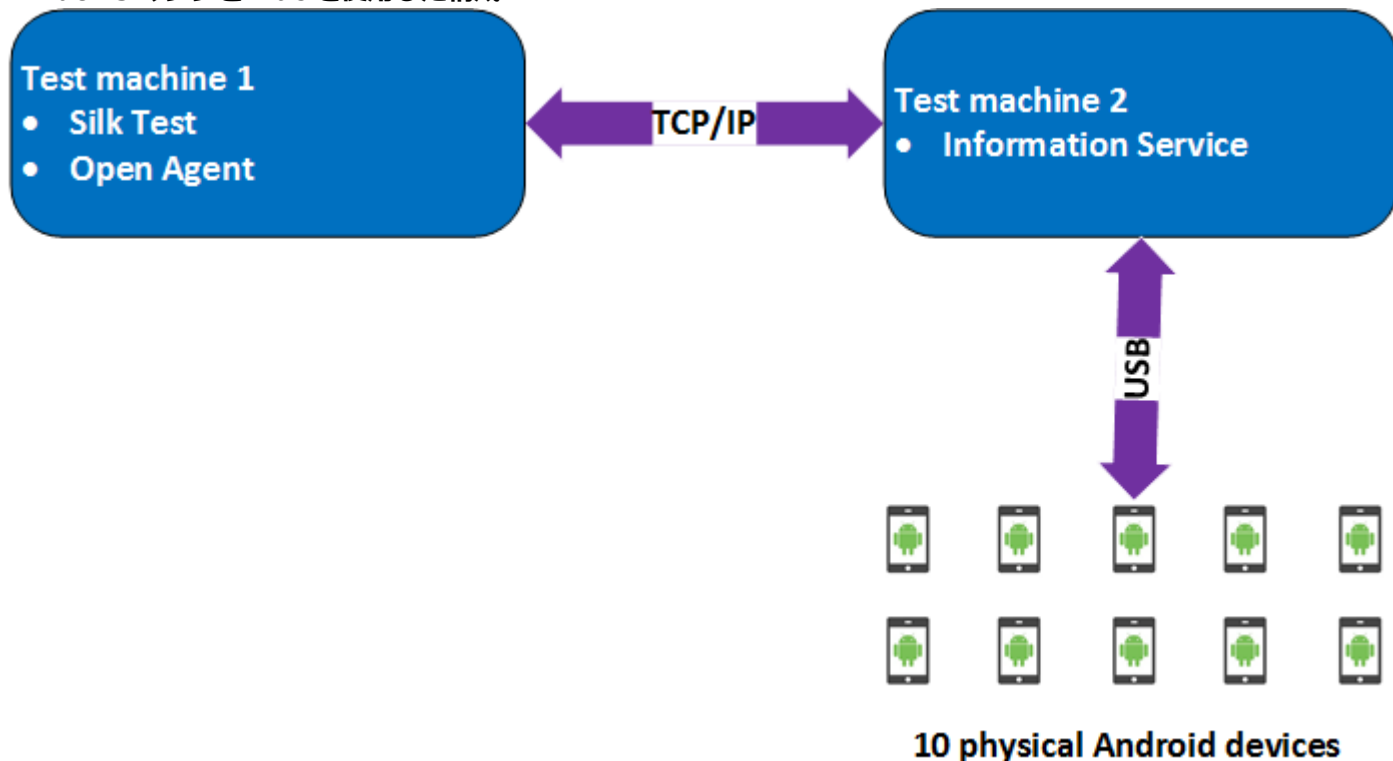
テストマシン1は、次のハードウェア仕様のLenovo ThinkPad T450です。

- Intel® Core™ i7 - 5600U CPU @ 2.60 GHz
- 2コア(4スレッド)
- 8 GB RAM

テストマシン 2 は、次のハードウェア仕様の Dell Precision T1700 です。

- Intel® Core™ i7 - 4770 CPU @ 3.40 GHz
- 4 コア (8 スレッド)
- 16 GB RAM

Windows マシンと Mac を使用した構成



さらに、2 台のテストマシンを使用したテストで、Windows マシンに Silk Test Workbench をインストールし、Mac には Silk Test Information Service をインストールして最初のマシンのリモート ロケーションとして設定しました。このような構成を使用して、最大 10 台の物理 Android デバイスまでの並列テストを行いました。

テストマシン 1 は、次のハードウェア仕様の Lenovo ThinkPad T450 です。

- Intel® Core™ i7 - 5600U CPU @ 2.60 GHz
- 2 コア (4 スレッド)
- 8 GB RAM

テストマシン 2 は、次のハードウェア仕様の Apple Mac Mini です。

- Intel® Core™ i5 - 4782U CPU @ 2.60 GHz
- 2 コア (4 スレッド)
- 16 GB RAM

iOS

Silk Test Workbench では、iOS デバイスまたは iOS シミュレータ上のモバイル アプリケーションをテストすることができます。

iOS の以前のバージョンと比べて、iOS 9.3 に対して大幅な変更が Apple によってなされたため、Silk Test は iOS 9.3 以降のモバイル アプリケーションのテストをサポートします。サポートされている iOS のバージョンのリストについては、『[リリースノート](#)』を参照してください。




注: iOS 11 上のモバイル アプリケーションのテストには Xcode 9 が必要です。Mac 上で Xcode 9 を使用している場合は、この Mac に接続 (または実行) されている iOS 11 より前のバージョンの iOS をインストールした物理デバイスおよびシミュレータ上でのテストはサポートされません。iOS 9.3 または iOS 10 の物理デバイスまたはシミュレータでテストする場合は、Xcode 8.3 を使用してください。

iOS 上のモバイル アプリケーションのテストにおける前提条件

iOS デバイスや iOS シミュレータ上のモバイル アプリケーションをテストする前に、次の前提条件を満たしていることを確認してください。


- 現在のバージョンの Silk Test Information Service が Mac 上にインストールされている。詳細については、「[Silk Test Information Service を Mac にインストールする](#)」を参照してください。
- 物理 iOS デバイス上でアプリケーションをテストする場合は、以下の項目を確認してください。
 - デバイスが Mac に接続されている。
 - デバイスが iOS のサポートされているバージョンである。サポートされている iOS のバージョンのリストについては、『[リリースノート](#)』を参照してください。
- iOS シミュレータ上でアプリケーションをテストする場合は、以下の項目を確認してください。
 - iOS シミュレータのイメージが Mac 上にインストールされている。
 - iOS シミュレータのイメージが iOS のサポートされているバージョンである。サポートされている iOS のバージョンのリストについては、『[リリースノート](#)』を参照してください。
- 物理 iOS デバイス上でアプリケーションをテストする場合は、デバイスと Mac の両方ともタイムゾーンが同じである。
- サポートするバージョンの Xcode が Mac 上にインストールされている。
- Windows マシン上に Silk Test Workbench がインストールされている。
- Mac が Windows マシンと同じネットワークにあり、Windows マシンにリモート ロケーションとして追加されている。
- iOS デバイス上でネイティブ モバイル アプリをテストする場合は、Developer Account でサインされたアプリの .ipa ファイルが作成されている。詳細については、「[iOS アプリのテストの準備](#)」を参照してください。
- iOS シミュレータ上でネイティブ モバイル アプリをテストする場合は、ZIP 形式に圧縮したアプリが作成されている。詳細については、「[iOS シミュレータ上のネイティブ モバイル アプリケーションのテスト](#)」を参照してください。
- iOS デバイスと iOS シミュレータの両方でネイティブ モバイル アプリをテストする場合は、サインした .ipa ファイルと ZIP した .app ディレクトリ (ファイル拡張子以外同じ名前) の両方が同じフォルダに作成されている。
- ネイティブ モバイル アプリをテストする場合は、iOS デバイスの ID がアプリをサインするのに使用した Developer Profile に関連付けられている。
- iOS デバイスのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、**設定 > 一般 > パスコードロック** を選択します。
- テスト中に Mac の画面がオフにならないようにしてください。オフになると、**再生ステータス** ダイアログ ボックスに何も表示されなくなります。
- iOS シミュレータ上でモバイル アプリケーションをテストする場合は、テスト中に Mac のディスプレイスリープが無効化されている。
- 物理 iOS デバイス上のネイティブ モバイル アプリをテストする場合は、デバイスで UI オートメーションが有効化されている。詳細については、「[iOS デバイスのテストの準備](#)」を参照してください。
- 物理 iOS デバイス上で Apple Safari を使用してのモバイル Web アプリケーションをテストする場合は、**Web インスペクタ** が有効化されている。詳細については、「[iOS デバイスのテストの準備](#)」を参照してください。
- Lightning コネクタを持つ iOS デバイスを使用することを Micro Focus では推奨しています。これは、Silk Test Workbench では、Lightning ケーブルで Mac に接続していない iOS デバイスのデバイス画面のライブ ビュー表示をサポートしていないためです。


物理 iOS デバイス上のネイティブ モバイル アプリケーションのテスト

 **注:** Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「ライセンス情報」を参照してください。


iOS 上のモバイル アプリケーションのテストにおける前提条件については、「[iOS 上のモバイル アプリケーションのテストにおける前提条件](#)」を参照してください。ネイティブ モバイル アプリケーションをテストする際の既知の制限については、「[モバイル ネイティブ アプリケーションのテストにおける制限事項](#)」を参照してください。

物理 iOS デバイス上のネイティブ モバイル アプリケーション (アプリ) やハイブリッド アプリケーションをテストするには、次のタスクを実行します。

1. テストする iOS デバイスを準備します。
詳細については、「[iOS デバイスのテストの準備](#)」を参照してください。
2. テストするアプリを準備します。
詳細については、「[iOS アプリのテストの準備](#)」を参照してください。
3. テストする Mac を準備します。詳細については、「[iOS 上でのモバイル アプリケーションのテストのための Mac の準備](#)」を参照してください。
4. iOS デバイスが接続されている Mac を、Silk Test がインストールされている Windows マシンに、リモート ロケーションとして追加します。
詳細については、「[リモート ロケーションの編集](#)」を参照してください。
 **注:** 任意の時点で、Mac に接続されている複数の物理 iOS デバイス上でテストできますが、iOS シミュレータは Mac 上で実行している 1 つに対してのみテストできます。Silk Test 17.5 Hotfix 1 以降を使用した場合、iOS 上のモバイル アプリケーションをテストするために、Mac 上で複数のユーザー セッションを使用する必要はありません。
5. モバイル アプリケーション用の Silk Test Workbench プロジェクトを作成します。
6. モバイル アプリケーション用のテストを作成します。
7. テストで実行する操作を記録します。記録 ウィンドウを開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。
8. **モバイル** タブを選択します。
9. アプリをテストするモバイル デバイスをリストから選択します。
10. ネイティブ モバイル アプリケーションを選択します。
 - モバイル デバイスまたはエミュレータ上にアプリをインストールする場合は、**参照** をクリックしてアプリ ファイルを選択するか、**アプリ ファイル** テキスト フィールドにアプリ ファイルへの完全パスを入力します。このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
 - iOS デバイス上に既にインストールされているアプリを使用する場合は、**バンドル ID** を指定します。
 - UFT Mobile 上で利用可能なアプリを使用する場合は、**アプリ識別子** を指定します。
11. **OK** をクリックします。
iOS デバイスやシミュレータのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、**設定 > Touch ID とパスコード** を選択します。
12. すべての操作の記録を終えたら、記録を停止します。
13. テストを再生します。
14. テスト結果を分析します。

 **注:** iOS デバイスと iOS シミュレータの両方でネイティブ モバイル アプリをテストする場合は、サインした .ipa ファイルと ZIP した .app ディレクトリ（ファイル拡張子以外同じ名前）の両方が同じフォルダに作成されている。

iOS シミュレータ上のネイティブ モバイル アプリケーションのテスト

 **注:** Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「ライセンス情報」を参照してください。


iOS 上のモバイル アプリケーションのテストにおける前提条件については、「[iOS 上のモバイル アプリケーションのテストにおける前提条件](#)」を参照してください。ネイティブ モバイル アプリケーションをテストする際の既知の制限については、「[モバイル ネイティブ アプリケーションのテストにおける制限事項](#)」を参照してください。

iOS シミュレータ上のネイティブ モバイル アプリケーション (アプリ) やハイブリッド アプリケーションをテストするには、次のタスクを実行します。

1. テストする Mac を準備します。詳細については、「[iOS 上でのモバイル アプリケーションのテストのための Mac の準備](#)」を参照してください。
2. アプリの Xcode プロジェクトで、iOS シミュレータ用にアプリをコンパイルします。
Xcode UI からでも、コマンドラインからでもアプリをコンパイルできます。たとえば、iOS 10.0 で iOS シミュレータ用のアプリをコマンドラインでコンパイルするには、次のコマンドを実行します。

```
xcodesbuild -sdk iphonesimulator10.0
```


3. アプリの .app ディレクトリを .zip ファイルに Zip します。
デフォルトでは、.app ディレクトリは、~/Library/Developer/Xcode/DerivedData ディレクトリにあります。Xcode で **File > Project Settings** をクリックすれば、ディレクトリがある場所を確認できます。
4. iOS シミュレータがインストールされている Mac を、Silk Test Workbench がインストールされている Windows マシンに、リモート ロケーションとして追加します。
詳細については、「[リモート ロケーションの編集](#)」を参照してください。

 **注:** Mac にインストールされている 1 つの iOS シミュレータでのみテストを実行できます。複数の Silk Test Workbench ユーザーが、同じ Mac にインストールされている複数の iOS シミュレータ上で同時にテストを実行することはできません。

5. モバイル アプリケーション用の Silk Test Workbench プロジェクトを作成します。
6. モバイル アプリケーション用のテストを作成します。
7. テストで実行する操作を記録します。記録 ウィンドウを開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。
8. **モバイル** タブを選択します。
9. リストから iOS シミュレータを選択します。
10. ネイティブ モバイル アプリケーションを選択します。
 - iOS シミュレータ上にアプリをインストールする場合は、**参照** をクリックして ZIP 形式のアプリ ファイルを選択するか、**アプリ ファイル** テキスト フィールドに ZIP 形式のアプリ ファイルへの完全パスを入力します。このパスでは、Silk Test Workbench は HTTP および UNC 形式をサポートします。
 - iOS シミュレータ上に既にインストールされているアプリを使用する場合は、**バンドル ID** を指定します。
11. **OK** をクリックします。
iOS デバイスやシミュレータのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、**設定 > Touch ID とパスコード** を選択します。
12. すべての操作の記録を終えたら、記録を停止します。

13テストを再生します。


14テスト結果を分析します。

 **注:** iOS デバイスと iOS シミュレータの両方でネイティブ モバイル アプリをテストする場合は、サインした .ipa ファイルと ZIP した .app ディレクトリ（ファイル拡張子以外同じ名前）の両方が同じフォルダに作成されている。

物理 iOS デバイス上のモバイル Web アプリケーションのテスト

iOS 上のモバイル アプリケーションのテストにおける前提条件については、「[iOS 上のモバイル アプリケーションのテストにおける前提条件](#)」を参照してください。モバイル Web アプリケーションをテストする際の既知の制限については、「[モバイル Web アプリケーションのテストにおける制限事項](#)」を参照してください。

物理 iOS デバイス上のモバイル Web アプリケーションをテストするには、次のタスクを実行します。


1. テストする iOS デバイスを準備します。
詳細については、「[iOS デバイスのテストの準備](#)」を参照してください。
2. テストする Mac を準備します。詳細については、「[iOS 上でのモバイル アプリケーションのテストのための Mac の準備](#)」を参照してください。
3. iOS デバイスが接続されている Mac を、Silk Test がインストールされている Windows マシンに、リモート ロケーションとして追加します。
詳細については、「[リモート ロケーションの編集](#)」を参照してください。
 **注:** 任意の時点で、Mac に接続されている複数の物理 iOS デバイス上でテストできますが、iOS シミュレータは Mac 上で実行している 1 つに対してのみテストできます。Silk Test 17.5 Hotfix 1 以降を使用した場合、iOS 上のモバイル アプリケーションをテストするために、Mac 上で複数のユーザー セッションを使用する必要はありません。
4. モバイル アプリケーション用の Silk Test Workbench プロジェクトを作成します。
5. モバイル アプリケーション用のテストを作成します。
6. テストで実行する操作を記録します。記録 ウィンドウを開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。
7. モバイル Web アプリケーションをテストするには：
 - a) **Web** タブを選択します。
 - b) 使用するモバイル ブラウザーを選択します。
 - c) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。
8. **OK** をクリックします。
iOS デバイスやシミュレータのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、**設定 > Touch ID とパスコード** を選択します。
9. すべての操作の記録を終えたら、記録を停止します。
- 10テストを再生します。
- 11テスト結果を分析します。

iOS シミュレータ上のモバイル Web アプリケーションのテスト

モバイル Web アプリケーションをテストする際の既知の制限については、「[モバイル Web アプリケーションのテストにおける制限事項](#)」を参照してください。

iOS シミュレータ上のモバイル Web アプリケーションをテストするには、次のタスクを実行します。

1. テストする Mac を準備します。詳細については、「[iOS 上でのモバイル アプリケーションのテストのための Mac の準備](#)」を参照してください。
2. iOS シミュレータがインストールされている Mac を、Silk Test がインストールされている Windows マシンに、リモート ロケーションとして追加します。
詳細については、「[リモート ロケーションの編集](#)」を参照してください。

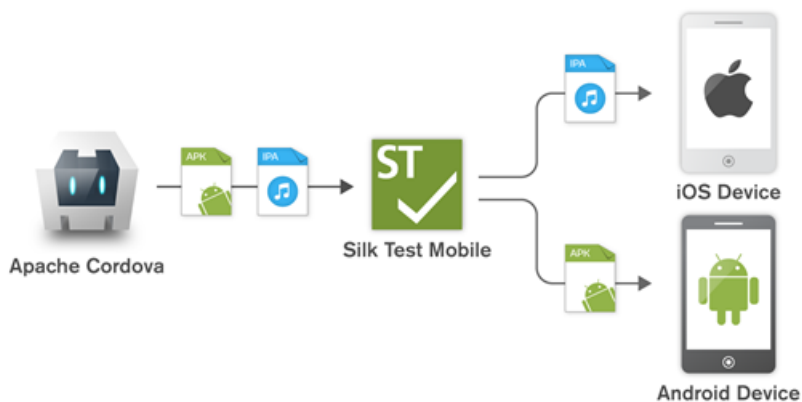
 **注:** 任意の時点で、Mac に接続されている複数の物理 iOS デバイス上でテストできますが、iOS シミュレータは Mac 上で実行している 1 つに対してのみテストできます。Silk Test 17.5 Hotfix 1 以降を使用した場合、iOS 上のモバイル アプリケーションをテストするために、Mac 上で複数のユーザー セッションを使用する必要はありません。

3. モバイル アプリケーション用の Silk Test Workbench プロジェクトを作成します。
4. モバイル アプリケーション用のテストを作成します。
5. テストで実行する操作を記録します。記録 ウィンドウを開始すると、**アプリケーションの選択** ダイアログ ボックスが開きます。
6. モバイル Web アプリケーションをテストするには：
 - a) **Web** タブを選択します。
 - b) 使用するモバイル ブラウザーを選択します。
 - c) **移動する URL の入力** テキスト ボックスに、開く Web ページを指定します。
7. **OK** をクリックします。
iOS デバイスやシミュレータのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、**設定 > Touch ID とパスコード** を選択します。
8. すべての操作の記録を終えたら、記録を停止します。
9. テストを再生します。
10. テスト結果を分析します。

iOS 上のハイブリッド アプリケーションのテスト

ハイブリッド アプリケーション (アプリ) は、デバイス上で実行されるネイティブ アプリケーションのようなアプリですが、HTML5、CSS、JavaScript などの Web テクノロジーを使用して記述されたアプリです。

Silk Test Workbench は、ネイティブ コンテナに埋め込まれた単一の Web ビューで構成されたデバッグ ハイブリッド アプリのテストにする完全なブラウザー サポートを提供します。このようなハイブリッド アプリの一般的な例は、Apache Cordova アプリケーションです。



リモート デバッグを有効化しない非デバッグ ハイブリッド アプリや、複数の Web ビューを含んだハイブリッド アプリをテストするには、Silk Test Workbench フォールバック サポートを有効化するために、オプション `OPT_ENABLE_MOBILE_WEBVIEW_FALLBACK_SUPPORT` を `TRUE` に設定します。詳細については、「詳細オプションの設定」を参照してください。フォールバック サポートを有効化すると、Silk Test Workbench は Web ビューのコントロールをブラウザー コントロールではなく、ネイティブ モバイル コントロールとして解決して処理します。たとえば、以下のコードは、ブラウザー サポートを使用したときのリンクのクリックです。

```
' VB .NET code
Agent.SetOption(Options.EnableMobileWebviewFallbackSupport, False)
_desktop.DomLink("//BrowserApplication//BrowserWindow//INPUT[@id='email']").Click()
```

フォールバック サポートを有効化すると、同じリンクをクリックするコードは次のようになります。

```
' VB .NET code
Agent.SetOption(Options.EnableMobileWebviewFallbackSupport, True)
```

```
desktop.Find("//BrowserApplication//BrowserWindow//MobileTextField[@resource-id='email']").Click()
```

iOS 上のハイブリッド アプリをテストする手順は、モバイル ネイティブ アプリケーションをテストする手順と同じです。詳細については、「[物理 iOS デバイス上のネイティブ モバイル アプリケーションのテスト](#)」または「[iOS シミュレータ上のネイティブ モバイル アプリケーションのテスト](#)」を参照してください。

iOS デバイス上のハイブリッド アプリをテストする前に、デバイスで **Web インспекタ** が有効化されていることを確認してください。詳細については、「[iOS デバイスのテストの準備](#)」を参照してください。

iOS デバイスのテストの準備



注: Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「[ライセンス情報](#)」を参照してください。

モバイル アプリケーションをテストするために iOS デバイスを準備するには：

1. Mac 上で Xcode を起動します。
2. iOS デバイスを Mac に接続します。
3. iOS デバイスで、**設定 > デベロッパ** をクリックします。



ヒント: デベロッパ メニューが iOS デバイスに表示されていない場合は、デバイスと Mac を再起動します。

4. **Enable UI Automation** をオンにします。
5. Apple Safari 上でモバイル Web アプリケーションをテストするには、**設定 > Safari > 詳細** をクリックします。
6. **Web インспекタ** をオンにします。
7. iOS シミュレータ上でテストを行う場合は、**Rotate Device Automatically** をオンにします。
この設定は、**Silk Test Configuration Assistant** を使うか、手動でオンにできます。Mac 上で **Configuration Assistant** を開くには、ステータス メニューの Silk Test アイコンをクリックして、**Configuration Assistant** を選択します。手動でオンにするには、以下の操作を実行します。
 - a) Mac 上で iOS シミュレータを開始します。
 - b) Xcode 9 以降を使用している場合は、**Hardware** メニューを開きます。
以前のバージョンの Xcode を使用している場合は、**Debug** メニューを開きます。
 - c) **Rotate Device Automatically** をオンにします。

iOS アプリのテストの準備

Silk Test Workbench を使用して特定の デバイス上で特定の iOS アプリをテストできるようにするには、次の項目を考慮する必要があります。

- 特定の デバイスに手動でインストールできる iOS アプリに対してのみテストを自動化できます。iOS アプリにサインできるようにするために、*Apple Developer Program* のメンバーシップに登録する必要があります。詳細については、「[メンバーシップの選択](#)」を参照してください。*Apple Developer Program* のメンバーシップに登録せずにテストを行う場合は、「[パーソナル チーム プロファイルを使用した物理 iOS デバイス上でのテスト](#)」を参照してください。



注: App Store で配布するように作成された アプリや、任意の iOS デバイスに手動でインストールできるアプリを自動的にテストできません。


- 特定の デバイスで アプリをインストールして実行する前に、iOS デバイスを Apple Developer アカウントを使用して登録する必要があります。
- デバイスに iOS アプリをインストールするには、Xcode を使用してアプリの IPA ファイルを作成する必要があります。特定の iOS デバイスでのテスト用に IPA ファイルを作成するには、Apple Developer Program のメンバーとして Xcode の *Archive* 機能を使用します。この機能には 2 つのオプションがあります。

- *Apple Developer Enterprise Program* のメンバーの場合、**Save for Ad Hoc Deployment** オプションを使用できます。
- *Apple Developer Enterprise Program* のメンバー以外の *Apple Developer Program* のメンバーの場合、**Save for Development Deployment** オプションを使用できます。


詳細については、「[Exporting Your App for Testing \(iOS, tvOS, watchOS\)](#)」を参照してください。

シミュレータ上の特定のアプリを Silk Test Workbench を使用してテストできるようにするには、Xcode を使用してアプリの Zip ファイルを作成してから iOS シミュレータ上にインストールします。詳細については、Xcode のドキュメントを参照してください。

Silk Test Information Service を Mac にインストールする


 **注:** Information Service を Mac にインストールするには、Mac の管理者権限が必要です。

Mac 上の Apple Safari や、iOS デバイスや Android デバイスを使ってテストを作成、実行するには、Mac 上に Silk Test Information Service (Information Service) をインストールします。Information Service をインストールして実行すると、Windows マシン上にインストールした Silk Test Workbench クライアントからテストを記録、再生できるようになります。

 **注:** Mac 上で記録することはできません。Mac をテスト ロケーションとして Silk Test Workbench に追加する場合は、Silk Test Workbench ドキュメントの「リモートロケーションの編集」を参照してください。

Information Service を Mac にインストールするには：

1. Silk Test Workbench ドキュメント「*Mac 上の Apple Safari を使用したテストにおける前提条件*」を参照してください。
2. Java JDK が Mac 上にインストールされていることを確認します。
3. iOS デバイス上でモバイルアプリケーションをテストする場合は、Xcode が Mac 上にインストールされていることを確認します。
4. Information Service セットアップ ファイル (SilkTestInformationService<バージョン>-<ビルド番号>.pkg) にアクセスします。
 - Silk Test のインストール時に Information Service セットアップ ファイルをダウンロードした場合は、Silk Test インストール ディレクトリ (C:\Program Files (x86)\Silk\SilkTest など) の macOS フォルダを開きます。
 - Silk Test のインストール時に Information Service セットアップ ファイルをダウンロードしなかった場合は、[Micro Focus SupportLine](#) からセットアップ ファイルをダウンロードできます。
5. SilkTestInformationService<バージョン>-<ビルド番号>.pkg ファイルを Mac にコピーします。
6. SilkTestInformationService<バージョン>-<ビルド番号>.pkg を実行して、Information Service をインストールします。
7. インストール ウィザードの指示に従います。
8. パスワードを尋ねられた場合、現在サインインしている Mac ユーザーのパスワードを入力します。
9. Apple Safari が開き、SafariDriver を信頼するかどうかを尋ねるメッセージ ボックスが表示されたら、**信頼** をクリックします。


 **注:** macOS 10.12 (Sierra) 以前上で Apple Safari 10 以前に対してテストを実行する場合は、Mac 上に SafariDriver をインストールする必要があります。リモート接続ではなく、直接 Mac にログインしている場合には、SafariDriver だけをインストールできます。

10. インストールを完了するために、現在の Mac ユーザーをログアウトします。Information Service が正しくインストールされていることを確認するには、Mac にログインします。

11. macOS Mojave (10.14) 以降の Mac に Information Service をインストールした場合、Mac にログインした後で Silk Test に対するオートメーションの承認を許可する必要があります。


許可を与える必要がある場合、Silk Test は要求の承認ダイアログを自動的に表示します。

- a) **OK** をクリックして、許可して情報ダイアログを閉じます。
- b) その後に表示されるすべての要求の承認ダイアログに対して、**OK** をクリックします。


 **重要:** これらの Silk Test に対する承認を許可しないと、Google Chrome 上の Web アプリケーションや iOS デバイスやシミュレーター上のモバイル アプリケーションを、この Mac でテストできなくなります。いずれかの承認ダイアログで、誤って **許可しない** をクリックしてしまった場合は、Mac 上でターミナルを開き、次のコマンドを実行してください。

```
sudo tccutil reset AppleEvents
```

その後、Mac を再起動して、再度承認ダイアログが表示されたら **OK** をクリックして許可します。**12**画面の右上にある Silk Test アイコンをクリックすると、利用可能なデバイスをブラウザーを確認できます。

 **ヒント:** Silk Test アイコンが表示されない場合は、Mac を再起動してください。

iOS 上でのモバイル アプリケーションのテストのための Mac の準備

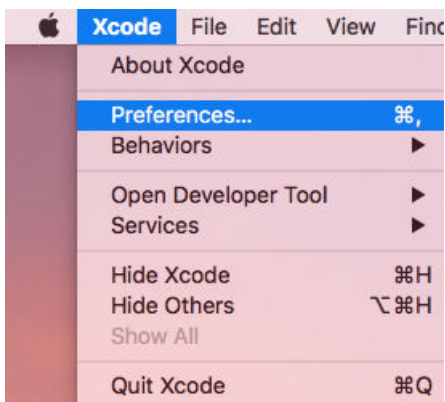
 **注:** Silk Test Workbench を使用してネイティブ モバイル アプリケーションやハイブリッド アプリケーションをテストするには、ネイティブ モバイル ライセンスが必要です。詳細については、「ライセンス情報」を参照してください。

iOS 上でモバイル アプリケーションをテストするためには、iOS デバイスを接続した、または iOS シミュレーターを実行している Mac が必要です。この Mac には Xcode がインストールされている必要があります。iOS 上のモバイル アプリケーションのテストにおける前提条件についての詳細は、「[iOS 上のモバイル アプリケーションのテストにおける前提条件](#)」を参照してください。

iOS テストを物理 iOS デバイス上で実行するには、**Silk Test Configuration Assistant** の指示に従って WebDriverAgentRunner Xcode プロジェクトを設定します。**Configuration Assistant** を開くには、ステータス メニューの Silk Test アイコンをクリックして、**Configuration Assistant** を選択します。

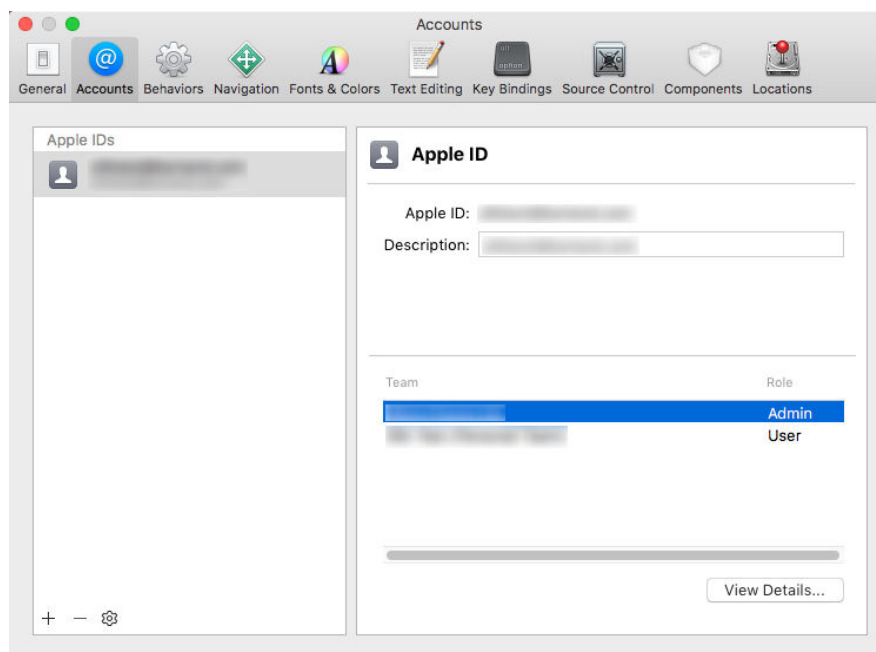
WebDriverAgentRunner Xcode プロジェクトを手動でビルドしたい場合には、次の手順を実行します。

1. Mac 上で Xcode を起動します。
2. **Xcode > Preferences** を選択します。




3. **Preferences** ウィンドウで、アカウントを選択します。

- a) **Accounts** タブを選択します。
- b) **Apple ID** を選択します。
- c) **Team** を選択します。
- d) **View Details** をクリックします。



4. **Apple Member Center** にアクセスし、開発チーム ID を取得します。
5. ターミナルで、`~/silk/silktest/conf/` に移動します。
6. `xcconfig` ファイル テンプレート `silktest.xcconfig.sample` を `silktest.xcconfig` という名前に変更します。
7. 開発チーム ID を `silktest.xcconfig` ファイルに追加します。
`DEVELOPMENT_TEAM = <your development team>`
8. Mac 上のターミナルで次のコマンドを実行すると、`WebDriverAgentRunner` プロジェクトが正しく準備できたかどうかを確認できます。
 - a) 物理 iOS デバイスの UDID (Unique Device ID) を見つけます。
`idevice_id -l`
 - b) `WebDriverAgentRunner` プロジェクトに移動します。
`cd /Application/Silk/Mobile/common/Appium/node_modules/appium-xcuitest-driver/WebDriverAgent`
 - c) `WebDriverAgent` がビルドできるかどうかをテストします。
`xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -xcconfig ~/.silk/silktest/conf/silktest.xcconfig -destination 'id=<udid>' test`
`<udid>` を先ほど見つけた UDID で置き換えます。

 **ヒント:** `xcodebuild` コマンドが失敗する場合には、エラーメッセージの手順に従ってください。さらに、`WebDriverAgentRunner` プロジェクトで Preferences ウィンドウを開き、**General** タブの **Automatically manage signing** チェックボックスがオフになっていることを確認します。
9. 省略可能 : `infoservice.properties` ファイルには、Silk Test Information Service のポートや、Mac 上でのすべてのテストの実行で使用されるケイパビリティを指定することができます。
 詳細については、「[Silk Test Information Service プロパティの編集](#)」を参照してください。

パーソナル チーム プロファイルを使用した物理 iOS デバイス上でのテスト

Apple Developer Program のメンバーシップに登録していない場合は、パーソナル チーム プロファイルを使用して、物理 iOS デバイス上のアプリケーションをテストできます。

1. Mac 上で、 /Application/Silk/Mobile/common/Appium/node_modules/appium-xcuitestdriver/ WebDriverAgent に移動します。
2. WebDriverAgent.xcodeproj プロジェクトを Xcode で開きます。
3. **TARGETS** リストから、WebDriverAgentLib ターゲットを選択します。
 - a) **General** タブをクリックします。
 - b) **Automatically manage signing** を選択します。
 - c) 開発チームを選択します。

Signing Certificate は自動的に選択されます。
4. **TARGETS** リストから、WebDriverAgentRunner ターゲットを選択します。
 - a) **General** タブをクリックします。
 - b) **Automatically manage signing** を選択します。
 - c) 開発チームを選択します。

Signing Certificate は自動的に選択されます。
5. WebDriverAgentRunner ターゲットに対するプロビジョニング プロファイルの作成に Xcode が失敗する場合は、ターゲットのバンドル ID を手動で変更します。
 - a) **Build Settings** タブをクリックします。
 - b) **Product Bundle Identifier** の値を Xcode が受け付ける適切な値に変更します。
 たとえば、**Product Bundle Identifier** が *com.facebook.WebDriverAgentRunner* の場合、*io.appium.WebDriverAgentRunner* や *io.borland.WebDriverAgentRunner* などに変更します。
 - c) **General** タブをクリックします。

これでターゲットにプロビジョニング プロファイルが設定されました。
6. WebDriverAgent.xcodeproj プロジェクトを保存します。
7. すべてが期待通り動作することを確認するために、ターミナルを開いてプロジェクトをビルドします。


```
xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination 'id=<udid>' test IPHONEOS_DEPLOYMENT_TARGET=10.3
```
8. WebDriverAgent アプリの再インストール時に問題が発生する場合は、同じプロビジョニング プロファイルを使ってデバイスに追加のアプリを永続的にインストールします。たとえば、WebDriverAgent Xcode プロジェクトの IntegrationApp をインストールします。
 - a) **TARGETS** リストから、IntegrationApp ターゲットを選択します。
 - b) **General** タブをクリックします。
 - c) **Automatically manage signing** を選択します。
 - d) 開発チームを選択します。
9. IntegrationApp ターゲットに対するプロビジョニング プロファイルの作成に Xcode が失敗する場合は、上記の WebDriverAgentRunner ターゲットに対する対処と同じ手順で、ターゲットのバンドル ID を手動で変更します。
- 10 IntegrationApp ターゲットの設定に成功したら、IntegrationApp を物理 iOS デバイスにインストールして実行します。
 - a) ターゲットと iOS デバイスを選択します。
 - b) **Play** をクリックします。

アプリのデバイスへのインストールが成功しても、次のようなエラー メッセージがコンソールや Appium ログ ファイルに出力される場合があります。

```
2017-01-24 09:02:18.358 xcodebuild[30385:339674] Error
Domain=com.apple.platform.iphones Code=-12 "Unable to launch
com.apple.test.WebDriverAgentRunner-Runner" UserInfo={NSLocalizedString=Unable to
launch com.apple.test.WebDriverAgentRunner-Runner, NSUnderlyingError=0x7fa839cadc60
{Error Domain=DTXMessage Code=1 "(null)" UserInfo={DTXExceptionKey=The operation
couldn't be completed. Unable to launch com.apple.test.WebDriverAgentRunner-Runner because
it has an invalid code signature, inadequate entitlements or its profile has not been explicitly
trusted by the user. : Failed to launch process with bundle identifier
```

```
'com.apple.test.WebDriverAgentRunner-Runner'}}} 2017-01-24 09:02:18.358
xcodebuild[30385:339674] Error Domain=IDETestOperationsObserverErrorDomain Code=5
"Early unexpected exit, operation never finished bootstrapping - no restart will be attempted"
UserInfo={NSLocalizedString=Early unexpected exit, operation never finished bootstrapping
- no restart will be attempted} Testing failed: Test target WebDriverAgentRunner encountered an
error (Early unexpected exit, operation never finished bootstrapping - no restart will be
attempted)
```

この問題は、開発者がそのデバイスで信頼されていないため発生します。そのデバイス上でアプリを手動で実行しようとする、**信頼されていない開発元** メッセージが表示される場合があります。

デバイス上でこの問題を解決するには、**設定 > 一般 > プロファイル** または **設定 > 一般 > デバイス管理** (デバイスの種類と iOS のバージョンによって異なります) に移動します。そして、開発元を信頼し、アプリの実行を許可します。

Silk Test Information Service プロパティの編集

infoservice.properties ファイルを使用して、Silk Test Information Service のポートや、HTTPS によるセキュアな接続を使用するかどうかを指定したり、Silk Test Information Service を実行しているマシン上で Silk Test がテストを実行するたびに毎回適用されるケイパビリティ (Capabilities) を指定できます。

- infoservice.properties.sample ファイルがあるディレクトリに移動します。
 - Windows マシン上で、%PROGRAMDATA%\Silk\SilkTest\conf に移動します (C:\ProgramData\Silk\SilkTest\conf など)。
 - macOS 上で、~/silk/silktest/conf/ に移動します。
- infoservice.properties.sample ファイルの名前を infoservice.properties に変更します。
- Silk Test Workbench と Information Service との通信に、HTTPS によるセキュアな接続を使用するかどうかを指定します。
 - HTTPS によるセキュアな通信を使用する場合は、infoservice.https.enabled を true に設定します。これは、デフォルトの設定です。
 - HTTPS によるセキュアな通信を使用しない場合は、infoservice.https.enabled を false に設定します。
- 省略可能 : HTTPS によるセキュアな通信を使用するように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、infoservice.default.https.port を設定します。
デフォルトの HTTPS ポートは 48561 です。ポート番号は、1 から 65535 の間の任意の数値を指定できます。
- 省略可能 : HTTPS によるセキュアな通信を使用するように指定した場合に、すべての HTTP リクエストを HTTPS ポートにリダイレクトするには、infoservice.http-to-https.enabled を true に設定します
デフォルト値は、false です。
- 省略可能 : HTTPS によるセキュアな通信を使用しないように指定した場合、Silk Test Workbench と Information Service の通信に未使用の別のポートを指定する場合には、infoservice.default.port を設定します。
デフォルトのポートは 22901 です。
- 省略可能 : Silk Test が HTTPS 接続で使用する証明書をユーザーの証明書で置き換える場合は、「[Information Service との HTTPS 接続に使用する証明書の置換](#)」を参照してください。
- ケイパビリティを指定するには、次の行を infoservice.properties ファイルに追加します。

```
customCapabilities=<custom_capability_1>;<custom_capability_2>;...
```

例：指定した言語での iOS シミュレータの実行

Mac 上で iOS シミュレータを常に同じ言語で実行するには、カスタム ケイパビリティ *language* および *locale* を指定します。たとえば日本語の場合、次の行を `infoservice.properties` ファイルに追加します。

```
customCapabilities=language=ja;locale=ja_JP
```

Silk Test Information Service を Mac からアンインストールする

Mac 上の Apple Safari に対するテストを実行する必要がなくなった場合など、次の手順で Silk Test Information Service を Mac からアンインストールすることができます。

1. Mac 上で、`uninstallInfoService.sh` のような新しいシェル ファイルを作成します。
2. 新しいファイルに以下のコードを入力します。

```
#!/bin/sh

if launchctl list | grep com.borland.infoservice ; then
  launchctl unload /Library/LaunchAgents/com.borland.infoservice.plist
  echo "unloading Launch Daemon"
fi

if [ -d "/Applications/Silk" ]
then
  sudo rm -rf /Applications/Silk
fi

if [ -f "/Library/LaunchAgents/com.borland.infoservice.plist" ]
then
  sudo rm /Library/LaunchAgents/com.borland.infoservice.plist
fi

if [ -f "/usr/local/bin/ideviceinstaller" ]
then
  sudo rm /usr/local/bin/ideviceinstaller
fi

exit 0
```

3. コマンドラインで `chmod +x uninstallInfoService.sh` を実行し、シェル ファイルの実行可能にします。
4. コマンドラインからシェル ファイルを実行します。

iOS デバイスの推奨設定

Silk Test Workbench を使用したテストを最適化するために、テストしたい iOS デバイスで次の設定を行ってください。

- 実際にユーザーが行った操作をテストに反映させるために、Apple Safari の自動入力とパスワードの保存を無効化します。設定 > Safari > パスワードと自動入力をタップし、ユーザー名とパスワード 設定をオフにします。
- iOS デバイスのテスト中に、スリープ モードに移行してはいけません。画面ロックとパスワードをオフにするには、設定 > 一般 > パスコードロック を選択します。

XCUITest を使用した iOS 上の既存のスク립トの実行

 **注目:** 以前のバージョンの Silk Test Workbench では、*Instruments* を使用して iOS デバイスを自動化していました。iOS 9.3 から、*Instruments* のサポートは *XCUITest* フレームワークのサポートに

Apple によって置き換えられたため、Silk Test Workbench でも *Instruments* をサポートできなくなりました。この変更により、既存の iOS テスト スクリプトを Silk Test Workbench の現在のバージョンで実行できなくなる場合があります。

- XCUITest における classname 属性の動作は Instruments での動作と異なります。ほとんどの場合、Silk Test Workbench はこの変更に対応します。しかし、このような classname 属性が原因で既存のテスト スクリプトが動作しなくなった場合には、対応するオブジェクトの新しいロケータを記録する必要があります。
- オブジェクトの階層が変更されました。

モバイル アプリケーションの記録

Silk Test Workbench とモバイル デバイスまたはエミュレータとの間の接続が一旦確立すると、デバイス上で実行する操作を記録できます。モバイル アプリケーションを記録するために、Silk Test Workbench では次の機能を持つ **記録** ウィンドウを使用します。

- テストするモバイル デバイスまたは Android エミュレータの画面を表示します。
- **記録** ウィンドウで操作を実行すると、モバイル デバイス上でも同じ操作が実行されます。
- 画面上のコントロールを操作すると、**記録** ウィンドウはデフォルトの操作を事前に選択します。
 - デフォルトの操作が Click の場合、コントロール上で左クリックすると、その操作が実行されます。右クリックすると、コントロールに対して利用可能な操作のリストが表示されます。この場合は、実行する操作を選択し、**OK** をクリックします。
 - デフォルトの操作が Click ではない場合、コントロールに対して有効なすべての操作がリストで表示されるので、実行したい操作を選択するか、単に **OK** をクリックして事前に選択された操作を受け入れます。

リストから操作を選択する場合、選択した操作のパラメータの値をパラメータ フィールドに入力することができます。Silk Test Workbench は自動的にパラメータを検証します。

- 記録中、Silk Test Workbench は記録ウィンドウの隣にマウスの位置を表示します。その表示をクリックすると、デバイス画面に絶対的な位置とアクティブ オブジェクトに相対的な位置を切り替えることができます。
- 記録を一時停止すると、画面上での操作は記録されないため、デバイスを記録を続けたい状態に変更することができます。
- 記録を停止すると、記録した操作でスクリプトが生成されるため、続いてテストの再生を行うことができます。

テストを再生するモバイル デバイスの選択

テストを再生するために使用するモバイル デバイスを定義できます。

- Silk Test Workbench の UI からテストを実行する場合、**モバイル デバイスの選択** ダイアログ ボックスが表示され、このダイアログ ボックスで選択したモバイル デバイス、Android エミュレータ、iOS シミュレータが使用され、テスト スクリプトで設定されているモバイル デバイス Silk Test Workbench は無視します。
- **モバイル デバイスの選択** ダイアログ ボックスが無効の場合 (**再び表示しない** チェック ボックスをオンにした場合)、個々のテスト スクリプトのアプリケーション構成によって、テストを実行するために使用するモバイル デバイスが決定されます。



注: モバイル デバイスの選択 ダイアログ ボックスを再び有効にするには、**ツール > オプション** をクリックして、**モバイル構成ダイアログの表示** オプションを **はい** に設定します。

- スクリプトをコマンド ラインや CI サーバーから実行する場合は、スクリプトのアプリケーション構成で接続文字列を指定します。

アプリケーション構成で指定したモバイル デバイスを上書きするには、mobiledevice パラメータを使用します。詳細については、STW.exe コマンド ラインの「パラメータ」トピックを参照してください。

- Silk Central からテストを実行する場合、接続文字列を指定する代わりに、Silk Central の実行定義の **配置** タブにある **モバイルデバイスの選択** 領域でモバイル デバイスを指定します。詳細については、『[Silk Central ヘルプ](#)』を参照してください。

デバイス プールなどがある場合に、特定のモバイル デバイスを指定したり、利用可能なデバイス群のサブセットをフィルタするために接続文字列を使用できます。最初に一致したデバイスが再生に使用されま
す。特に指定がない場合には、次のルールに従って一致したモバイル デバイスが使用されます（高い優先
度順）。

- リモート ロケーションに接続されたモバイル デバイスよりも、ローカル マシンに接続されたモバイル デバイスが優先されます。
- ブラウザーの種類が接続文字列で指定されている場合、古いバージョンのブラウザよりも、新しいバージョンのブラウザが優先されます。
- 古いプラットフォームよりも、新しいプラットフォームが優先されます。
- 物理デバイスがエミュレータやシミュレータよりも優先されます。
- アルファベット順で後者のデバイス名のデバイスが優先されます。たとえば、"iphone 5"という名前のデバイスよりも、"iphone 6"という名前のデバイスが優先されます。

例：リモート マシンに接続されている Android デバイス上のアプリに対する接続文字列

リモート マシンに接続されている Android デバイス上で MyApp.apk アプリをテストするには、接続文字列は次のようになります。

```
"platformName=Android;deviceName=MotoG3;host=http://10.0.0.1;app=MyApp.apk"
```

例：Mac の iOS シミュレータ上のアプリに対する接続文字列

```
"platformName=iOS;platformVersion=10.0;deviceName=iPhone6;host=10.0.0.1;app=MyApp.ipa;isSimulator=true"
```

Silk Test Workbench から UFT Mobile デバイスの直接使用

Micro Focus UFT Mobile は、実デバイスとエミュレーターでのエンドツーエンドの品質ラボを提供し、モバイル アプリのテスト、監視、最適化を支援して、ユーザー エクスペリエンスの向上に貢献します。



注: Silk Test Workbench では、UFT Mobile 3.0 以降によって管理されているデバイスに対するテストをサポートします。

UFT Mobile によって管理されているデバイスにアクセスする方法には、Silk Test Workbench から直接アクセスする方法と、Silk Central 経由で行う方法があります。


UFT Mobile によって管理されているデバイスに Silk Test Workbench から直接アクセスするには：

1. UFT Mobile をリモート ロケーションとして追加します。
詳細については、『[リモート ロケーションの編集](#)』を参照してください。
2. iOS 上でテストする場合は、次の IPA ファイルがサインされていることを確認します。
 - HP4M-Agent.ipa
 - HPMC-AgentLauncher.ipa
 - WebDriverAgentRunner-Runner.ipa



注: Silk Test Workbench は、UFT Mobile を介した iOS シミュレータのテストをサポートしません。

以上により、**アプリケーションの選択** ダイアログに UFT Mobile デバイスが表示されます。テストするデバイスを選択してください。

 **注:** Silk Test Mobile と UFT Mobile の両方で同時に同じデバイスをテストできません。製品を切り替えて使用する場合には、自動テストで使用していたすべてのアプリを削除し、モバイル デバイスを再起動する必要があります。

Android Silk Test Mobile を使って Android デバイスをテストする前に、または UFT Mobile をインストールする前に、デバイスから次のアプリを削除する必要があります。


- Appium Android Input Manager
- Appium Settings
- io.appium.uiautomator2.server
- io.appium.uiautomator2.server.text
- MC Agent
- Silk Screencast
- Unlock


iOS Silk Test Mobile を使って iOS デバイスをテストする前に、または UFT Mobile をインストールする前に、デバイスから WebDriverAgent を削除する必要があります。


UFT Mobile は、これらのアプリを再インストールする可能性があります。

UFT Mobile によって管理されているデバイス上で、CI サーバーやコマンド ラインからテストを再生したい場合は、リモート接続を構成する代わりに、デバイスの接続文字列を指定できます。接続文字列は次のようになります。

```
"deviceName=MotoG3;platformName=Android;host=http://<UFT Mobile server>:8080;hostType=MC;userName=<UFT Mobile user name>;password=<UFT Mobile password>"
```


 **注:** セキュリティ上の理由から、Silk Test Workbench で新しいアプリケーション構成を作成すると、UFT Mobile パスワードは暗号化されて保存されます。接続文字列では、この暗号化されたパスワードを使用することを Micro Focus は推奨します。

 **注:** UFT Mobile 上で管理されているデバイスをテストする場合は、Silk Test Workbench は TypeKeys メソッドや SetText メソッドを使用して **Enter** などのキー コードの入力をサポートしません。さらに、Silk Test Workbench は iOS デバイス上の **ホーム** ボタンの押下をサポートしません。

 **注:** Android エミュレータ上でテストする場合は、GPU HW アクセラレーションを無効にしてください。

Silk Central 経由での UFT Mobile デバイスの使用

Micro Focus UFT Mobile は、実デバイスとエミュレーターでのエンドツーエンドの品質ラボを提供し、モバイル アプリのテスト、監視、最適化を支援して、ユーザー エクスペリエンスの向上に貢献します。

 **注:** Silk Test Workbench では、UFT Mobile 3.0 以降によって管理されているデバイスに対するテストをサポートします。


UFT Mobile によって管理されているデバイスにアクセスする方法には、Silk Test Workbench から直接アクセスする方法と、Silk Central 経由で行う方法があります。

UFT Mobile によって管理されているデバイスに Silk Central 経由でアクセスするには：

1. Silk Test Workbench と Silk Central を統合します。

詳細については、「*Silk Test Workbench と Silk Central の統合*」を参照してください。


2. UFT Mobile を使用するように Silk Central を設定します。

 **注:** UFT Mobile のインストール時に、適切なバージョンの Android SDK が使用されていることを確認してください。Silk Test Workbench で同じバージョンを使用するように設定するには、環境


変数 `SILK_ANDROID_HOME` に `C:\Users\<ユーザー>\AppData\Local\Android\android-sdk` などを指定します。詳細については、『*Silk Central* ヘルプ』を参照してください。

3. iOS 上でテストする場合は、次の IPA ファイルがサインされていることを確認します。

- HP4M-Agent.ipa
- HPMC-AgentLauncher.ipa
- WebDriverAgentRunner-Runner.ipa

 **注:** Silk Test Workbench は、UFT Mobile を介した iOS シミュレータのテストをサポートしません。

以上により、**アプリケーションの選択** ダイアログに UFT Mobile デバイスが表示されます。テストするデバイスを選択してください。


 **注:** Silk Test Mobile と UFT Mobile の両方で同時に同じデバイスをテストできません。製品を切り替えて使用する場合には、自動テストで使用していたすべてのアプリを削除し、モバイル デバイスを再起動する必要があります。


Android Silk Test Mobile を使って Android デバイスをテストする前に、または UFT Mobile をインストールする前に、デバイスから次のアプリを削除する必要があります。

- Appium Android Input Manager
- Appium Settings
- io.appium.uiautomator2.server
- io.appium.uiautomator2.server.text
- MC Agent
- Silk Screencast
- Unlock

iOS Silk Test Mobile を使って iOS デバイスをテストする前に、または UFT Mobile をインストールする前に、デバイスから WebDriverAgent を削除する必要があります。

UFT Mobile は、これらのアプリを再インストールする可能性があります。

 **注:** UFT Mobile 上で管理されているデバイスをテストする場合は、Silk Test Workbench は TypeKeys メソッドや SetText メソッドを使用して **Enter** などのキー コードの入力をサポートしません。さらに、Silk Test Workbench は iOS デバイス上の **ホーム** ボタンの押下をサポートしません。

 **注:** Android エミュレータ上でテストする場合は、GPU HW アクセラレーションを無効にしてください。

UFT Mobile との HTTPS 接続用証明書のインストール

Silk Test Workbench と UFT Mobile 間の通信に HTTPS によるセキュアな通信を使用するには、UFT Mobile 用の証明書を Open Agent が使用する Java のインストール ディレクトリにインストールする必要があります。

1. ブラウザーを開きます。
2. セキュアな UFT Mobile サーバーの HTTPS URL に移動します。
3. セキュアな UFT Mobile サーバーの証明書をエクスポートします。

証明書をエクスポートする方法については、使用しているブラウザのドキュメントを参照してください。

たとえば、Mozilla Firefox を使用している場合は、次の操作を実行します。

- a) 安全な接続でないというメッセージを Mozilla Firefox が表示した場合、**詳細** をクリックします。
- b) **例外の追加...** をクリックします。
- c) **セキュリティ例外の追加** ダイアログで、**表示** をクリックします。
- d) **証明書ビューアー** で、**詳細** タブを選択します。

- e) **エクスポート** をクリックします。
 - f) **証明書をファイルに保存** ダイアログで、証明書ファイルを保存する場所に移動します。
 - g) 証明書の名前を指定します (mc.crt など)。
 - h) **保存** をクリックします。
4. 証明書ファイルを保存した場所を開きます。
 5. 証明書ファイルをダブルクリックして、ウィザードに従って証明書をインストールします。
 - a) **証明書をすべて次のストアに配置する** を選択してから、**参照** をクリックします。
 - b) **信頼されたルート証明機関** を選択して、ウィザードを完了します。
 6. セキュリティ警告が表示されたら、インストールを確認して **はい** をクリックします。
 7. Open Agent を再起動します。
 8. UFT Mobile サーバーまたはコネクタ マシンの URL をブラウザ上で更新し、SSL 証明書関連のエラーが表示されないことを確認します。

UFT Mobile パスワードの変更

UFT Mobile パスワードが変更された場合、Silk Test Workbench の **ロケーションの編集** ダイアログを使って、新しいパスワードを適用する必要があります。

1. **ツール > リモート ロケーションの編集** をクリックします。 **リモート ロケーション** ダイアログ ボックスが表示されます。
2. UFT Mobile リモート ロケーションを選択し、**編集** をクリックします。 **ロケーションの編集 - UFT Mobile** ダイアログ ボックスが表示されます。
3. **パスワード** フィールドに新しいパスワードを入力します。
4. 省略可能 : **テスト** をクリックして、新しいパスワードが正しく機能することを確認します。
5. 複数の接続文字列で同じ UFT Mobile を使用している場合は、**コピー** をクリックして暗号化したパスワードをクリップボードにコピーします。そして、コピーした暗号化したパスワードを接続文字列に貼り付けることができます。
6. **OK** をクリックします。

Sauce Labs デバイスの使用

Sauce Labs は、自動テスト プラットフォームを提供しており、さまざまなモバイル デバイスやモバイル プラットフォームのバージョン上でテストを行うことができます。自分自身のインフラストラクチャとしてデバイスを購入、保守する必要はありません。

Sauce Labs デバイスには Silk Central を介してアクセスすることができます。次の手順に従って設定してください。

1. Silk Test Workbench が Silk Central と統合されていることを確認してください。
詳細については、「*Silk Test Workbench と Silk Central の統合*」を参照してください。
2. Silk Central が Sauce Labs を使用するように設定されていることを確認してください。
詳細については、*Silk Central* ヘルプ を参照してください。

以上により、**アプリケーションの選択** ダイアログに Sauce Labs デバイスが表示されます。テストするデバイスを選択してください。

モバイル デバイスの接続文字列

接続文字列 は、テストに使用するモバイル デバイスを指定します。モバイル テストを実行する場合、Silk Test Workbench は接続文字列を使用してモバイル デバイスに接続します。接続文字列は、アプリケーション構成の主要な一部です。テスト対象アプリケーションを構成するときに、接続文字列は設定されます。接続文字列を変更するには、**アプリケーション構成の編集** ダイアログ ボックスを使用します。



注: Silk Central からテストを実行する場合、接続文字列を指定する代わりに、Silk Central の実行定義の **配置** タブにある **モバイルデバイスの選択** 領域でモバイル デバイスを指定します。詳細については、『[Silk Central ヘルプ](#)』を参照してください。

デバイス プールなどがある場合に、特定のモバイル デバイスを指定したり、利用可能なデバイス群のサブセットをフィルタするために接続文字列を使用できます。最初に一致したデバイスが再生に使用されません。特に指定がない場合には、次のルールに従って一致したモバイル デバイスが使用されます（高い優先度順）。

- リモート ロケーションに接続されたモバイル デバイスよりも、ローカル マシンに接続されたモバイル デバイスが優先されます。
- ブラウザーの種類が接続文字列で指定されている場合、古いバージョンのブラウザよりも、新しいバージョンのブラウザが優先されます。
- 古いプラットフォームよりも、新しいプラットフォームが優先されます。
- 物理デバイスがエミュレータやシミュレータよりも優先されます。
- アルファベット順で後者のデバイス名のデバイスが優先されます。たとえば、"iphone 5" という名前のデバイスよりも、"iphone 6" という名前のデバイスが優先されます。

次のコンポーネントが接続文字列で使用できます。

コンポーネント	説明
deviceName	モバイル デバイスの名前。物理モバイル デバイス上でテストをする場合、デバイス ID を代わりに使用します。ワイルドカードをサポートします。大文字と小文字は区別されません。
platformName	Android または iOS。必須。
deviceId	省略可能：モバイル デバイスの ID。物理モバイル デバイスでテストをする場合に、デバイス名の代わりに使用します。ワイルドカードをサポートします。大文字と小文字は区別されません。
platformVersion	省略可能：Android または iOS のバージョン。特定の Android または iOS のバージョンのモバイル デバイス上でのみテストする場合に、バージョンを指定します。ワイルドカードをサポートします。大文字と小文字は区別されません。
browserVersion	省略可能：特定のブラウザのバージョンでのみテストする場合に、ブラウザの種類と共に使用します。ワイルドカードをサポートします。大文字と小文字は区別されません。
host	省略可能：設定しない場合は、任意のリモート ロケーションがホストとして使用されます。ワイルドカードをサポートします。大文字と小文字は区別されません。
app	モバイル デバイスにまだインストールされていないアプリへのフルパス。たとえば、app=MyApp.apk です。
<ul style="list-style-type: none"> • appPackage • appActivity 	Android デバイスにインストールされているアプリをテストする場合は、使用するアプリのパッケージとアクティビティの両方を指定します。たとえば、appPackage=silktest.insurancemobile;appActivity=.LoginActivity です。
bundleId	iOS デバイスにインストールされているアプリのバンドルの識別子。たとえば、app=MyApp.ipa または bundleId=silktest.InsuranceMobile などです。
mobileCenterAppIdentifier	UFT Mobile によって管理されているモバイル デバイスにインストールされているアプリの識別子。たとえば、mobileCenterAppIdentifier=com.microfocus.silktest.testapp です。
mobileCenterAppUploadNumber	UFT Mobile によって管理されているモバイル デバイスにインストールされているアプリのアップロード番号。複数回アップロードされたアプリケーションを一意に識別するために使用します。この番号が指定されていない場合は、Silk Test Workbench は最新のアップロードを使用します。たとえば、mobileCenterAppUploadNumber=3 です。
noReset	省略可能：ネイティブ モバイル アプリケーションをテストする場合に設定できます。app が指定されている場合にのみ有効です。テストの前にアプリを再インストールしない場合は True。テストの前にアプリを再インストールする場合は False を指定します。デフォルト値は、False です。

コンポーネント	説明
isSimulator	省略可能: iOS シミュレータ上でのみテストを実行する場合に指定します。デバイス名を代わりに使用できます。
isPhysicalDevice	省略可能: 物理デバイス上でのみテストを実行する場合に指定します。デバイス名を代わりに使用できます。

デバイス プールを使用して、テストで実際に使用されるデバイスを確認するには、MobileDevice クラスの GenerateConnectionString メソッドの戻り値を使用できます。

モバイル デバイスまたは Android エミュレータ上のモバイル Web アプリケーションのテスト

モバイル デバイスまたは Android エミュレータ上でモバイル Web アプリケーションをテストする場合、接続文字列は次の要素から構成されます。

1. モバイル デバイス名 (MotoG3 など)、またはデバイス ID (11111111 など)。



注: デバイス ID は可読性に欠けるため、デバイス名がユニークであれば、デバイス名を接続文字列に使用することを Micro Focus では推奨します。

2. プラットフォーム名。
3. ブラウザーのバージョン。ブラウザーの種類の設定との組み合わせでのみ使用されます。
4. 特定のリモート マシンの IP アドレスまたはホスト名 (10.0.0.1 など)。**リモート ロケーションの編集** ダイアログ ボックスで指定したリモート ロケーションの名前をホスト名として使用することもできます (MyRemoteLocation など)。リモート ロケーション名を使用する場合に、ワイルドカードを使用することもできます。ローカル マシンに接続されている Android デバイスをテストする場合は、ローカル マシンの IP アドレスまたはホスト名を指定します。

例: 利用可能な任意の Android デバイスの接続文字列

```
"platformName=Android"
```

例: ローカル マシンに接続されている Android デバイス上のブラウザーに対する接続文字列

ローカル マシンに接続されている Android デバイス上でモバイル ブラウザーをテストするには、接続文字列は次のようになります。

```
"deviceName=MotoG3;platformName=Android;host=localhost"
```

または

```
"platformName=Android;deviceId=11111111;host=localhost"
```

例: リモート マシンに接続されている Android デバイス上のブラウザーに対する接続文字列

リモート Android デバイス上のモバイル ブラウザーをテストするには、接続文字列は次のようになります。

```
"deviceName=MotoG3;platformName=Android;host=10.0.0.1"
```

```
"deviceName=MotoG3;platformName=Android;host=MyRemoteLocation*"
```

例：UFT Mobile によって管理されている Android デバイスの接続文字列

UFT Mobile によって管理されているデバイス上で、CI サーバーやコマンドラインからテストを再生したい場合は、リモート接続を構成する代わりに、デバイスの接続文字列を指定できます。接続文字列は次のようになります。

```
"deviceName=MotoG3;platformName=Android;host=http://<UFT Mobile server>:8080;hostType=MC;userName=<UFT Mobile user name>;password=<UFT Mobile password>"
```



注：セキュリティ上の理由から、Silk Test Workbench で新しいアプリケーション構成を作成すると、UFT Mobile パスワードは暗号化されて保存されます。接続文字列では、この暗号化されたパスワードを使用することを Micro Focus は推奨します。

例：Mac に接続されている iOS デバイス上のブラウザーに対する接続文字列

リモート iOS デバイス上のモバイル ブラウザーをテストするには、接続文字列は次のようになります。

```
"deviceName=myiPhone6;platformName=iOS;host=10.0.0.1"
```

モバイル デバイスまたは Android エミュレータ上のネイティブ モバイル アプリケーションのテスト

モバイル デバイスまたは Android エミュレータ上でネイティブ モバイル アプリケーションをテストする場合、接続文字列は次の要素から構成されます。

1. モバイル デバイス名 (MotoG3 など)、またはデバイス ID (11111111 など)。



注：デバイス ID は可読性に欠けるため、デバイス名がユニークであれば、デバイス名を接続文字列に使用することを Micro Focus では推奨します。

2. プラットフォーム名。
3. 特定のリモート マシンの IP アドレスまたはホスト名 (10.0.0.1 など)。**リモート ロケーションの編集** ダイアログ ボックスで指定したリモート ロケーションの名前をホスト名として使用することもできます (*MyRemoteLocation* など)。リモート ロケーション名を使用する場合に、ワイルドカードを使用することもできます。ローカル マシンに接続されている Android デバイスをテストする場合は、ローカル マシンの IP アドレスまたはホスト名を指定します。
4. テストするアプリのファイルの名前または、ファイルが Web サーバー上にある場合には、ファイルの URL。たとえば、C:/MyApp.apk や MyApp.ipa など。
 - Android アプリは、常に .apk ファイルを指定します。
 - 物理デバイス上の iOS アプリは、常に .ipa ファイルを指定します。
 - シミュレータ上の iOS アプリは、ZIP ファイルまたは、*app* という名前のディレクトリを指定します。

例：リモート マシンに接続されている Android デバイス上のアプリに対する接続文字列

リモート マシンに接続されている Android デバイス上で MyApp.apk アプリをテストするには、接続文字列は次のようになります。

```
"platformName=Android;deviceName=MotoG3;host=http://10.0.0.1;app=MyApp.apk"
```

例 : Mac に接続されている iOS デバイス上のアプリに対する接続文字列

リモートマシンに接続されている iOS デバイス上で MyApp.ipa アプリをテストするには、接続文字列は次のようになります。

```
"platformName=iOS;deviceName=MyiPhone;host=http://10.0.0.1;app=MyApp.ipa"
```

iOS シミュレータ上のモバイル Web アプリケーションのテスト

iOS シミュレータ上でモバイル Web アプリケーションをテストする場合、接続文字列は次の要素から構成されます。

1. プラットフォーム名 (iOS)。
2. プラットフォームのバージョン (10.0 など)。
3. モバイルデバイス名 (iPhone6 など)。
4. iOS シミュレータを実行している Mac の IP アドレスまたはホスト名。

例 : Mac の iOS シミュレータ上のブラウザーに対する接続文字列

```
"platformName=iOS;platformVersion=10.0;deviceName=iPhone6;host=10.0.0.1;isSimulator=true"
```

iOS シミュレータ上のネイティブ モバイル アプリケーションのテスト

Mac の iOS シミュレータ上でネイティブ モバイル アプリケーションをテストする場合、接続文字列は次の要素から構成されます。

1. プラットフォーム名 (iOS)。
2. プラットフォームのバージョン (10.0 など)。
3. モバイルデバイス名 (iPhone6 など)。
4. リモートマシンの IP アドレスまたはホスト名 (10.0.0.1 など)。
5. テストするアプリの名前 (MyApp.ipa など)。

例 : Mac の iOS シミュレータ上のアプリに対する接続文字列

```
"platformName=iOS;platformVersion=10.0;deviceName=iPhone6;host=10.0.0.1;app=MyApp.ipa;isSimulator=true"
```

モバイルデバイスの操作


モバイルデバイス进行操作したり、テスト対象アプリケーションでスワイプのような操作を実行するには、次の手順を実行します。

1. **記録** ウィンドウで、**モバイルデバイス操作の表示** をクリックします。モバイルデバイスに対して実行できるすべての操作がリストされます。
2. リストからリストから実行したい操作を選択します。
3. Android デバイスまたはエミュレータで、スワイプを記録するには、マウスの左ボタンを押したままマウスを動かします。
4. テストの記録を続行します。

モバイルデバイスの開放

モバイルデバイスに対するテストを記録または再生する場合、Open Agent インスタンスはそのデバイスの所有権を確保します。これによって、Open Agent は、他の Silk Test ユーザーがそのデバイスを使用す

ることを防止します。デバイス上でのテストの記録や再生が終了した後に、他の Silk Test ユーザーがデバイスを使用できるようにするために、Silk Test クライアントが閉じたとき、無人のテスト プロセスが完了したとき、または Open Agent が閉じたときに、Silk Test は自動的にデバイスを開放します。また、デバイスを手動で開放することもできます。


 **注:** モバイルデバイスを開放すると、モバイル デバイス上のテスト対象アプリケーション (AUT) は閉じられます。

記録後のモバイルデバイスの開放

他の Silk Test ユーザーがモバイル デバイスでテストできるようにするために、記録後にデバイスを開放します。

記録が完了した後にモバイル デバイスを開放するには、次のいずれかを実行します。

- システム トレイから Open Agent を停止する。
- Silk Test Workbench を閉じる。並列テストが有効な場合には、この操作でのみデバイスが開放されます。
- UFT Mobile によって管理されているデバイスをテストしている場合は、UFT Mobile UI を使ってデバイスを開放することも可能です。詳細については、UFT Mobile のドキュメントを参照してください。

 **注:** モバイルデバイスを開放すると、モバイル デバイス上のテスト対象アプリケーション (AUT) は閉じられます。

再生後のモバイルデバイスの開放

他の Silk Test ユーザーがモバイル デバイスでテストできるようにするために、再生後にデバイスを開放します。

再生が完了した後に手動でモバイル デバイスを開放するために、次のいずれかを実行することもできます。

- モバイル Web アプリケーションのテストを再生した場合、BrowserApplication クラスの Close メソッド、または CloseSynchron メソッド を使用します。これらのメソッドの詳細については、API ドキュメントを参照してください。

```
'VB .NET code
WebBrowser.Close()
```

- モバイル ネイティブ アプリケーションのテストを再生した場合、MobileDevice クラスの CloseApp メソッドを使用します。


たとえば、次のように入力します。

```
'VB .NET code
Dim mobileDevice = _desktop.MobileDevice()
mobileDevice.CloseApp()
```

- VB .NET テストの再生後にデバイスを開放するには、Agent.DetachAll() ステートメントをテスト スクリプトに追加します。

次の条件を満たしている場合は、モバイル デバイスは自動的に開放されます。

- Open Agent を閉じる。
- 無人テスト中にテスト プロセスが停止する。並列テストが有効な場合には、この操作でのみデバイスが開放されます。
- Silk Test Workbench を閉じる。並列テストが有効な場合には、この操作でのみデバイスが開放されます。

 **注:** モバイルデバイスを開放すると、モバイル デバイス上のテスト対象アプリケーション (AUT) は閉じられます。

モバイル アプリケーションのテストにおける setLocation メソッドの使用

- iOS 上でのネイティブ モバイル アプリケーションのテスト時に、SetLocation メソッドはサポートされません。
- Android 6.0 より前のバージョンの Android 上でのネイティブ モバイル アプリケーションのテスト時に、SetLocation メソッドを使用する場合は、**擬似ロケーションを許可** を有効にする必要があります。設定は、Android デバイスまたはエミュレータの設定を開き、**開発者向けオプション** をタップします。
- Android 6.0 以降上でのネイティブ モバイル アプリケーションのテスト時に、SetLocation メソッドを使用する場合は、**Appium Settings** をアプリとして設定する必要があります。設定は、Android デバイスまたはエミュレータの設定を開き、**開発者向けオプション** > **仮の現在地情報アプリを選択** をタップします。そして、**Appium Settings** を選択します。



注: Appium Settings という項目は、Android デバイスまたはエミュレータ上の Appium でテストを既に実行した場合にのみ表示されます。

- Windows マシン上の Android エミュレーターでテストするとき、SetLocation メソッドを使用するには：
 1. エミュレーターを実行している Windows マシン上で、%userprofile% フォルダーに移動します。
 2. .emulator_console_auth_token ファイルの内容を削除します。このファイルは、エミュレーターを最初に起動したときに作成されます。
 3. .emulator_console_auth_token ファイルを読み取り専用を設定します。

Silk Central によって管理されているデバイスの使用

モバイル テストを管理し、構成テストを実行するために、Silk Central を使用できます。

Silk Test Workbench と Silk Central を統合します。

Silk Test Workbench と Silk Central の統合についての詳細は、「*Silk Test Workbench と Silk Central の統合*」を参照してください。

Silk Central の特定のユーザーに対して構成されたデバイス プロバイダーが、**アプリケーションの選択**、**ブラウザーの選択**、**モバイル デバイスの選択** ダイアログ ボックスで、リモート ブラウザーやモバイル デバイスのリストに表示されます。Silk Test Workbench は、Silk Central によって管理されているデバイスを、その名前と、Silk Central で構成されたデバイス プロバイダーのホスト名/URL と共に表示します。

モバイル アプリケーションのテスト時のトラブルシューティング

[アプリケーションの選択] ダイアログにモバイル デバイスが表示されない理由

Silk Test Workbench がモバイル デバイスやエミュレータを認識できないと、**アプリケーションの選択** ダイアログの **モバイル** タブにはデバイスやエミュレータが表示されません。さらに、**アプリケーションの選択** ダイアログの **Web** タブにも、そのデバイスやエミュレータ上にインストールされているモバイル ブラウザーが表示されません。

Silk Test Workbench が、次の何れかが原因でモバイル デバイスまたはエミュレータを認識していない可能性があります。

原因	解決策
エミュレータが実行されていない。	エミュレータを開始します。

原因	解決策
Android Debug Bridge (adb) がモバイル デバイスを認識しない。	<p>モバイル デバイスが adb によって認識されているかどうか確認するには：</p> <ol style="list-style-type: none"> 1. Android SDK をインストールしたフォルダで、Android Debug Bridge (adb) がある場所に移動します。Android SDK がインストールされていない場合、C:\Program Files (x86)\Silk\SilkTest\ng\Mobile\windows\AndroidTools\platform-tools に移動して、Silk Test Workbench がインストールした adb を使用します。 2. Shift を押しながら、ファイル エクスプローラ ウィンドウで右クリックします。 3. コマンド ウィンドウをここで開く を選択します。 4. コマンド ウィンドウで、adb devices を入力して、アタッチしたすべてのデバイスのリストを得ます。 5. デバイスがリストされない場合、USB デバッグがデバイスで有効化されていること、および適切な USB ドライバがインストールされていることを確認します。 6. 「adb server is out of date」のようなエラーが発生する場合は、C:\Program Files (x86)\Silk\SilkTest\ng\Mobile\windows\AndroidTools\platform-tools の adb のバージョンがローカルの Android SDK の adb のバージョンと一致していることを確認してください。詳細については、「<i>Open Agent</i> とデバイスとの接続が不安定な場合の対処方法」を参照してください。
デバイスのオペレーティング システムのバージョンを Silk Test Workbench がサポートしていない。	サポートするモバイル オペレーティング システムのバージョンについては、 リリース ノート を参照してください。
デバイスの USB ドライバがローカル マシンにインストールされていない。	デバイスの USB ドライバをローカル マシンにインストールしてください。詳細については、「 <i>USB ドライバのインストール</i> 」を参照してください。
USB デバッグがデバイスで有効化されていない。	USB デバッグをデバイスで有効化してください。詳細については、「 <i>USB デバッグの有効化</i> 」を参照してください。



注: どの解決策を適用しても解決できない場合は、デバイスを再起動してみてください。

URL に移動せずに Silk Test Workbench が Chrome for Android で URL を検索する理由

アドレス バーに入力された URL を、Chrome for Android が検索として解釈する場合があります。回避策として、URL に移動するコマンドをスクリプトに手動で追加できます。

adb サーバーが正しく起動しない場合にすべきこと

Android Debug Bridge (adb) サーバーが開始するとき、ローカル TCP ポート 5037 にバインドし、adb クライアントから送信されてくるコマンドをリッスンします。すべての adb クライアントは、ポート 5037 を使用して、adb サーバーと通信します。adb サーバーは、5555 から 5585 の範囲 (エミュレータやデバイスで使用される範囲) で奇数のポートをスキャンしてエミュレータやデバイス インスタンスを探します。adb はこれらのポートの変更を許しません。adb 開始中に問題が発生した場合、これらの範囲のポートの 1 つが、他のプログラムによって既に使用されているかどうか確認します。

詳細については、<http://developer.android.com/tools/help/adb.html> を参照してください。

Open Agent とデバイスとの接続が不安定な場合の対処方法

Android SDK、または Android Debug Bridge (adb) を使用するその他のツールをインストールしている場合、Silk Test Workbench が使用する adb サーバー以外のサーバーが実行中の可能性があります。バージョンの異なる adb サーバーが実行中の場合、Open Agent とデバイスとの接続が不安定になったり、接続できない場合があります。

このようなバージョンの不一致によるエラーを避けるには、環境変数 `SILK_ANDROID_HOME` に Android SDK ディレクトリへのパス (C:\%Users%\<ユーザー>%AppData%\Local%\Android%\android-sdk など) を指定してください。Information Service が実行中の場合、Windows のサービス マネージャーを使用して、Silk Test Information Service を再起動して更新した環境変数を適用する必要があります。環境変数が設定されていない場合は、Silk Test Workbench は Silk Test Workbench に同梱されたバージョンの adb を使用します。

エラー「メモリの割り当てに失敗しました : 8」が発生する理由

エミュレータを開始しているときに、システムが十分なメモリを割り当てることができない場合に、このエラーが表示されます。以下を行ってみてください。

1. エミュレータのメモリ オプションの RAM サイズを下げる
2. Intel HAXM の RAM サイズを下げる RAM サイズを下げるには、IntelHaxm.exe を再度実行して、**Change** を選択します。
3. **タスク マネージャ** を開き、十分なフリー メモリが利用可能かどうかを確認します。不足している場合、プログラムを閉じてメモリを開放してください。

iOS デバイスのテスト時に「Silk Test は指定したアプリを開始できません」というエラーが発生する理由

このエラーが発生する原因として、以下の理由が考えられます。

原因	解決策
iOS デバイスがデベロッパ モードになっていない。	次の 2 種類の方法のいずれかで、デベロッパ モードを有効化できます。 <ul style="list-style-type: none">• Xcode がインストールされている Mac にデバイスを接続し、テストするアプリをデバイスで開始します。• プロビジョニング プロファイルをデバイスに追加します。<ol style="list-style-type: none">1. Xcode を開きます。2. Window > Devices を選択します。3. iOS デバイスを右クリックします。4. Show Provisioning Profiles を選択します。5. プロビジョニング プロファイルを追加します。
デバイスの iOS のバージョンを最近更新した。	<ol style="list-style-type: none">1. Xcode を開きます。2. Window > Devices を選択します。3. Xcode がシンボル ファイルを処理するまで待機します。
UI オートメーションが iOS デバイスで有効化されていない。	<ol style="list-style-type: none">1. 設定 > デベロッパ を選択します。2. Enable UI Automation をオンにします。
Web インスペクタ が iOS デバイスで有効化されていない (モバイル Web アプリケーションのテストの場合)。	<ol style="list-style-type: none">1. 設定 > Safari > 詳細 をクリックします。

原因	解決策
	2. Web インспекタ をオンにします。
テストするアプリがテストしようとしている iOS デバイスの iOS バージョン用にビルドされていない。	Xcode を使用してデバイスの iOS バージョン用にアプリをビルドします。
ソフトウェア・アップデート ダイアログ ボックスが iOS デバイス上で開いている。	ダイアログ ボックスを閉じ、ソフトウェアの自動アップデートを無効化します。 1. 設定 > iTunes & App Store > 自動ダウンロード を選択します。 2. アップデート をオフにします。

Android デバイスの動的ハードウェア コントロールに戻るボタンだけが表示される理由

テストの開始時に Android デバイスや Android エミュレータの画面がロックされると、デバイスやエミュレータが動的ハードウェア コントロールに **戻る** ボタンだけを表示する場合があります。

この問題を解決するには、Open Agent を停止し、デバイスを再起動してから、デバイスの設定を画面のロックをしないように設定してください。

Android デバイスまたはエミュレータにキーボードが表示されなくなる理由

Unicode 文字列をサポートするために、Silk Test Workbench は標準キーボードをカスタム キーボードに置き換えます。そして、テストの完了時に元のキーボードに戻します。テスト中にエラーが発生すると、カスタム キーボードが設定されたまま、元に戻らない場合があります。

この問題を解決するには、**設定 > 言語と入力 > 現在のキーボード** を開き、手動で元のキーボードに戻してください。

テスト中にデバイスが応答しなくなる理由

テストの開始時に、デバイス、エミュレータ、シミュレータの画面がロックされると、Silk Test Workbench は画面のロックを解除できず、デバイス、エミュレータ、シミュレータが操作に応答しなくなる場合があります。

この問題を解決するには、Open Agent を停止し、デバイスの設定を画面のロックをしないように設定してください。

Information Service を Mac にインストールできない理由

システム環境設定の **セキュリティとプライバシー** で、**一般** タブの **ダウンロードしたアプリケーションの実行許可** 設定が **Mac App Store と確認済みの開発元からのアプリケーションを許可** (デフォルト値) に設定されている場合、Information Service セットアップを開いているときに次のエラー メッセージが表示されます。

"SilkTestInformationService<バージョン>.pkg" は、開発元が未確認のため開けません。

この問題を解決するには、次のいずれかを行います。

- セットアップ ファイルを右クリックして、**開く** を選択します。警告メッセージが表示されても、ファイルを開くことができます。
- **ダウンロードしたアプリケーションの実行許可** 設定を **すべてのアプリケーションを許可** に設定します。
- ファイルを開いた後、システム環境設定の **セキュリティとプライバシー** の **一般** タブを開き、**このまま開く** をクリックします。

Android アプリの記録時に記録ウィンドウが真っ暗になる理由

金融取引を処理するアプリなど、高レベルなセキュリティを必要とする Android アプリでは、Silk Test Workbench がアプリのキャプチャをできないようにするために、**FLAG_SECURE** フラグが設定されてい

る可能性があります。Silk Test Workbench は、記録時に Android デバイスのスクリーンショットやビデオを利用しますが、テストする Android アプリにこのフラグが設定されていると、**記録中** ウィンドウにはデバイスの真っ黒な画面が表示されます。Silk Test でこのようなアプリをテストするには、テスト中に FLAG_SECURE フラグを設定しないよう、アプリの開発チームに依頼してください。

Android エミュレータでのテスト時に Silk Test Workbench がビデオを表示しない理由

エミュレータがレンダリングにコンピュータのグラフィックカードを使用している場合、Silk Test Workbench のビデオキャプチャが機能しない場合があります。この問題を解決するには、ソフトウェアでグラフィックでエミュレートします。

1. **Android Virtual Device Manager** を開きます。
2. エミュレータの **Actions** 列の **Edit** をクリックします。
3. **Virtual Device Configuration** ダイアログ **Emulated Performance** 領域で、リストから **Software** を選択します。

クラウド環境でのテスト時に Silk Test Workbench がビデオを表示しない理由

クラウド環境でテストする場合、必要なポートがオープンになっていないことなどが原因で、テストの記録や再生時にビデオの表示が機能しない場合があります。

この問題を解決するには、`infoservice.properties` ファイルに WebDriver ホストの URL リストを指定します。このプロパティファイルについての詳細は、「*Silk Test Information Service* プロパティの編集」を参照してください。`infoservice.disableScreencastHosts` オプションをファイルに追加し、次のように入力します。

```
infoservice.disableScreencastHosts=<URL_1>,<URL_2>, ...
```

例：

```
infoservice.disableScreencastHosts=http://my-webdriver-server-url.com:80/wd/hub
```

ワイルドカードとしてアスタリスク (*) を使用して、`*my-webdriver-server-url.com` のような URL パターンを指定することができます。

この設定により、指定したホストでの記録と再生時に、Silk Test Workbench はビデオの代わりに連続したスクリーンショットを表示するようになります。

Xcode のインストールバージョンを変更する方法

Xcode の最新版にアップグレードしてしまった場合など、使用している Xcode のバージョンを Silk Test Workbench がサポートしていない場合、iOS でテストする際にエラーメッセージが表示される場合があります。

インストールした Xcode のバージョンをサポートするバージョンで置換するには、サポートするバージョンの Xcode を <https://developer.apple.com/download/more/> からダウンロードし、サポートされていないバージョンをダウンロードしたバージョンで置換します。サポートする Xcode のバージョンについての情報は、『[リリースノート](#)』を参照してください。

Mac のディスクの空き容量がなくなった場合の対処

Silk Test Workbench は iOS デバイスの自動化に Instruments を使用します。このツールは、`/Library/Caches/com.apple.dt.instruments` ディレクトリに大きなログファイルを生成するため、Mac のディスクの空き容量を圧迫している場合があります。この問題を解決するために、手動であるいは cron ジョブを使用して、これらのログファイルを定期的に削除することを Micro Focus は推奨します。たとえば、毎日同じ時間にファイルを削除するには、次の手順を実行します。

1. ターミナルで「`sudo crontab -e`」を入力します。crontab を root として編集できるエディタが開きます。
2. 次の行を crontab に追加します。

```
0 2 1 * * find /Library/Caches/com.apple.dt.instruments -mtime +10 -delete
```

3. crontab を保存します。

この例では、10 日より前のすべてのログ ファイルが、毎日午前 2 時にディレクトリから削除されます。

エラー メッセージ「WebDriver エージェントにテスト用の署名ができません」によってテストが失敗する理由

物理 iOS デバイス上でテストを行う場合、通常このエラーは WebDriverAgent アプリのビルド プロセス中に、プロビジョニング プロファイルでサインされていないか問題があったことを意味します。

デバイスが接続されている Mac マシンで次のコマンドを実行して、実際の問題を確認できます。

```
cd /Applications/Silk/Mobile/common/Appium/node_modules/appium-xcuitest-driver
xcodebuild -project WebDriverAgent.xcodeproj -scheme WebDriverAgentRunner -destination
'id=<udid>' test
```

/Applications/Silk/Mobile/common/Appium/node_modules/appium-xcuitest-driver フォルダにある Resources フォルダが存在し、そのフォルダに WebDriverAgent.bundle ファイルがあることを確認します。存在しない場合は、このフォルダを作成し、空の WebDriverAgent.bundle ファイルを作成します。たとえば、次のコマンドを実行します。

```
mkdir -p Resources/WebDriverAgent.bundle
```

Developer Tools Access による他のプロセスの制御の要求を抑える方法

iOS 上でテストの実行を開始すると、つぎのメッセージのようなメッセージ ボックスが表示される場合があります。

Developer Tools Access は、デバッグを続けるためにほかのプロセスを制御する必要があります。これを許可するには、パスワードを入力してください。パスワードを入力して許可します。

このメッセージを抑制するには、ターミナルで次のコマンドを実行します。

```
sudo /usr/sbin/DevToolsSecurity --enable
```

iPad 上のモバイル Web アプリケーションのテスト時に矩形領域がずれる理由

iPad 上のモバイル Web アプリケーションのテスト時にコントロールを囲む矩形領域がずれている場合、複数のブラウザ タブが開いており、タブ バーが表示されている場合があります。この問題を回避するには、1 つを残して、ほかのすべてのタブを閉じてください。

Silk Test Workbench のアップデート後にデバイス上でテストの記録と再生が動作しない理由

Silk Test Workbench の新しいバージョンにアップデートすると、Silk Test Workbench の以前のバージョンでモバイル テスト用に使用してきた物理モバイル デバイス上の Appium アプリも自動的にアップデートされます。何らかの理由でこれらのアプリが自動的にアップデートされないと、そのデバイス上でテストの記録と再生が正常に動作しない場合があります。

Silk Test Workbench のアップデート後に、Android デバイスでこのような問題が発生した場合は、次のアプリをデバイスから手動でアンインストールしてください。

- Appium Android Input Manager
- Appium Settings
- io.appium.uiautomator2.server
- io.appium.uiautomator2.server.test
- Unlock

Silk Test Workbench のアップデート後に、iOS デバイスでこのような問題が発生した場合は、WebDriverAgentRunner をデバイスから手動でアンインストールしてください。

モバイル アプリケーションを記録できない理由

Silk Test Workbench では、Appium を使ってモバイル アプリケーションをテストします。Appium でネットワーク プロキシ設定が設定されていると、Silk Test Workbench の記録を妨害する場合があります。モバイル デバイスまたはエミュレータのネットワーク プロキシ設定を無効にしてみてください。

Android デバイス上でテストを実行できない理由

一部の Android デバイスでは、Silk Test Workbench などを使ったデバイス上のモバイル アプリケーションのテストができないよう、あらかじめ設定されている場合があります。たとえば、Xiaomi Mi Mix 2 の前提条件は他のデバイスと同様ですが、さらに次の設定が必要になります。Xiaomi Mi Mix 2 でテストを実行する前に、次の手順を実行してください。

1. デバイスの開発者モードを有効にします。
2. **設定 > その他の設定 > 開発者向けオプション** に移動します。
3. **USB デバッグ** を有効にします。
4. **スリープモードにしない** を有効にします。
5. **USB 経由でインストールする** を有効にします。
6. **USB デバッグ** を有効にします。
7. **MIUI の最適化をオンにする** を無効にします。

iOS13 上でのハイブリッド アプリのテスト実行が iOS 12 に比べて遅くなった理由

iOS 13 以降では UIWebView コントロールが廃止されたため、テスト結果が不安定になりました。UIWebView を WKWebView に移行してください。

テストの再生に Chrome for Android を使用する方法

デフォルトでは、**ブラウザーの選択** ダイアログ ボックスを使用して、再生に使用するブラウザーを選択できます。

スクリプトをコマンド ラインや CI サーバーから実行する場合は、スクリプトのアプリケーション構成で接続文字列を指定できます。アプリケーション構成で指定したブラウザーを上書きするには、`silktest.configurationName` 環境変数を使用します。

BrowserApplication クラスの `browsertype` プロパティを使用して、再生に使用するブラウザーの種類を設定することもできます。ただし、`browsertype` は Chrome for Android の明示的な値を含みません。

テストを再生するブラウザーとして Chrome for Android を使用するように指定するには、`browsertype` に `GoogleChrome` を設定し、Android をプラットフォームとして指定します。Android が指定されると、デスクトップ マシン上の Google Chrome の代わりに Chrome for Android を使用して、Silk Test Workbench はテストを実行します。

使用例

次のサンプル コードは、`silktest.configurationName` を使用して Nexus 7 上の Chrome for Android を使用するテストの基本状態を設定する方法を示しています。

```
SET
silktest.configurationName="platformName=Android;deviceName=Nexus
7;host=10.0.0.1 - Chrome"
```

次の のサンプル コードは、`browsertype` を使用して Chrome for Android を使用するテストの基本状態を設定する方法を示しています。

```
'VB .NET code
Dim baseState = New BrowserBaseState(BrowserType.GoogleChrome,
"demo.borland.com/InsuranceWebExtJS/")
baseState.SetConnectionString("platformName=Android")
baseState.Execute()
```

モバイル Web アプリケーションのテストにおける制限事項

モバイル ブラウザ上でのテストの再生とロケータの記録のサポートは、サポートされている他のブラウザほど完全なものではありません。モバイル Web アプリケーションに対するテストの再生とロケータの記録の既知の制限事項を以下に示します。

- 次のクラス、インターフェイス、メソッド、プロパティは、モバイル Web アプリケーションでは現時点ではサポートされません。
 - BrowserApplication クラス。
 - CloseOtherTabs メソッド
 - CloseTab メソッド
 - ExistsTab メソッド
 - GetActiveTab メソッド
 - GetSelectedTab メソッド
 - GetSelectedTabIndex メソッド
 - GetSelectedTabName メソッド
 - GetTabCount メソッド
 - ImageClick メソッド
 - OpenTab メソッド
 - SelectTab メソッド
 - DomElement クラス。
 - DomDoubleClick メソッド
 - DomMouseMove メソッド
 - GetDomAttributeList メソッド
 - IKeyable インターフェイス。
 - PressKeys メソッド
 - ReleaseKeys メソッド
- Silk Test Workbench は、iOS 上の Apple Safari を使用した HTML フレームおよび iframe のテストをサポートしません (HTML フレームおよび iframe 上のテキスト解決を含む)。

テキスト解決は次のメソッドを含みます。

- TextCapture
- TextClick
- TextExists
- TextRectangle
- 横固定モードでの記録はシステム バーに仮想ボタンを含むエミュレータに対してサポートされません。このようなエミュレータは、回転を正しく検出せずに、横固定モードのシステム バーを画面の下部ではなく画面の右側に配置します。ただし、このようなエミュレータは縦固定モードで記録することができます。
- モバイル アプリケーションに対する XPath 式では、HTML DOM の HTML 属性だけがサポートされません。Silk Test Workbench は、XPath 式のプロパティをサポートしません。
- Android 上でのモバイル Web アプリケーションのテストでは、Silk Test Workbench は、拡大縮小をサポートしません。
- BrowserWindow クラスの以下の JavaScript 警告処理メソッドが、Original Android Stock (AOSP) ブラウザー上でのテストでは機能しません。
 - AcceptAlert メソッド
 - DismissAlert メソッド

- GetAlertText メソッド
- IsAlertPresent メソッド
- 任意の時点で、Mac に接続されている複数の物理 iOS デバイス上でテストできますが、iOS シミュレータは Mac 上で実行している 1 つに対してのみテストできます。
- モバイル Web アプリケーションのテストを開始する前に、ブラウザーのタブが開いていないことを確認してください。



ヒント: iPad 上で Apple Safari のタブを無効に出来ます。設定 > Safari を選択して、**タブバーを表示** をオフにすると無効になります。

- モバイル Web アプリケーションのテスト中に、ブラウザーのタブは 1 つだけ開くことができます。
- Silk Test Workbench は、ネイティブ モバイル アプリケーションによって開かれたモバイル Web アプリケーションのテストをサポートしません。
- iOS 上での Web アプリケーションおよびハイブリッド アプリケーションのテストには、iOS 11 以降のバージョンが必要です。

ネイティブ モバイル アプリケーションのテストにおける制限事項

ネイティブ モバイル アプリケーションに対するテストの再生とロケータの記録の既知の制限事項は次の通りです。

- 次のクラス、インターフェイス、メソッド、プロパティは、ネイティブ モバイル アプリケーションでは現時点ではサポートされません。
 - IKeyable インターフェイス。
 - PressKeys メソッド
 - ReleaseKeys メソッド
 - MobileDevice クラス。
 - iOS 上でのネイティブ モバイル アプリケーションのテスト時に、SetLocation メソッドはサポートされません。
 - Android 6.0 より前のバージョンの Android 上でのネイティブ モバイル アプリケーションのテスト時に、SetLocation メソッドを使用する場合は、**擬似ロケーションを許可** を有効にする必要があります。設定は、Android デバイスまたはエミュレータの設定を開き、**開発者向けオプション** をタップします。
 - Android 6.0 以降上でのネイティブ モバイル アプリケーションのテスト時に、SetLocation メソッドを使用する場合は、**Appium Settings** をアプリとして設定する必要があります。設定は、Android デバイスまたはエミュレータの設定を開き、**開発者向けオプション** > **仮の現在地情報アプリを選択** をタップします。そして、**Appium Settings** を選択します。



注: **Appium Settings** という項目は、Android デバイスまたはエミュレータ上の Appium でテストを既に実行した場合にのみ表示されます。

- iOS 上でのテスト時に、XCUIElementTypeSwitch クラスの getValue メソッドがチェック状態に応じて、文字列 0、1 ではなく、文字列 false、true を返します。
- 横固定モードでの記録はシステム バーに仮想ボタンを持つ Android エミュレータではサポートされません。このようなエミュレータは、回転を正しく検出せずに、横固定モードのシステム バーを画面の下部ではなく画面の右側に配置します。ただし、このようなエミュレータは縦固定モードで記録することができます。
- モバイル アプリケーションに対する XPath 式では、HTML DOM の HTML 属性だけがサポートされます。Silk Test Workbench は、XPath 式のプロパティをサポートしません。
- 任意の時点で、Mac に接続されている複数の物理 iOS デバイス上でテストできますが、iOS シミュレータは Mac 上で実行している 1 つに対してのみテストできます。Silk Test 17.5 Hotfix 1 以降を使用した場合、iOS 上のモバイル アプリケーションをテストするために、Mac 上で複数のユーザー セッションを使用する必要はありません。

- Silk Test Workbench は、Android と iOS の両方とも、ネイティブ モバイル アプリケーションのテスト時にテキスト解決をサポートしません。

テキスト解決は次のメソッドを含みます。

- TextCapture
- TextClick
- TextExists
- TextRectangle
- Silk Test Workbench は、複数の Web ビューを持つネイティブ モバイル アプリケーションのテストをサポートしません。
- iOS 上でのテスト時に、isVisible プロパティの状態は、要素が非表示であったとしても常に true になります。
- iOS 上でのテスト時に、複数のステップを持つスワイプ操作は、あるポイントへスワイプし、マウス ポインタをリリースした後、次のポイントへスワイプします。iOS の以前のバージョンでは、この操作はスワイプ間でマウス ポインタをリリースしません。
- iOS 上でのテスト時に、Silk Test Workbench はピンチ以外のマルチタッチ操作をサポートしません。
- iOS 上でのテスト時に、Silk Test Workbench は PinchIn メソッドをサポートしません。
- iOS 上でのテスト時に、警告ダイアログ ボックスに対して承認と解除のみを行えます。**キャンセル** ボタンは利用できないため、Silk Test Workbench はダイアログを解除できません。デフォルトの操作はダイアログの承認です。
- Android 上でのテスト時に、Silk Test Workbench は [Animation](#) クラスのコントロールに対する自動同期を行いません。
- Android 上でのトーストのテスト時には、次の制限事項があります。
 - 記録時に、トースト用の矩形領域は、トーストの実際の位置とは関係なく、Silk Test Workbench の **記録** ウィンドウの下部に表示されます。
 - 記録時と再生時に、トーストがすぐに表示された場合でも、Silk Test Workbench によるトーストの検出に、常に 5 秒間かかります。
- iOS 上でのテスト時に、Silk Test Workbench は UIView.animate 関数または UIView.animateWithDuration 関数を呼び出すコントロールに対する自動同期を行いません。

アプリケーション デリゲートでアニメーションの速度を速くすることで、問題が回避できる可能性があります。

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
    //...
    if NSProcessInfo.processInfo().environment["automationName"] == "Silk Test" {
        // Speed animations up (recommended)
        window!.layer.speed = 100;
    }
}
```

このようなアニメーションを完全に無効にすることは、アプリケーションの挙動が変わってしまう可能性があるため、Micro Focus では推奨していません。しかし、アニメーションの速度を速めても同期問題が解決できない場合は、次のようにしてアプリケーション デリゲートでアニメーションを完全に無効化することもできます。

```
func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
    //...
    if NSProcessInfo.processInfo().environment["automationName"] == "Silk Test" {
        UIView.setAnimationsEnabled(false)
    }
}
```

- iOS 上でのテスト時には、さらに次の制限事項があります。

- テストの記録および再生時にパフォーマンスの低下を感じる場合があります。
- iOS の内部的な変更の影響で一部のコントロールのロケータが変ったため、既存のテストが動作しない場合があります。
- フォーカスを持たないテキスト フィールドが、テキスト フィールドとして解決されない場合があります。テキスト フィールドを正しく解決するために、操作する前にテキスト フィールドのクリックなどを実行して、テキスト フィールドにフォーカスを移してください。

ネイティブ モバイル アプリに対するメソッドの動的呼び出し

動的呼び出しを使うと、モバイル ネイティブ アプリに対して Appium WebDriver のメソッドを直接呼び出すことができます。これは、Appium WebDriver のメソッドが Silk Test Workbench API を介して公開されていない場合に有用です。



注: 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。

オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

サポートされているメソッド

- Android 上のネイティブ モバイル アプリのテストでは、[Appium Java-client API](#) の AndroidDriver クラスで利用可能なメソッドを Silk Test Workbench はサポートします。
- iOS 上のネイティブ モバイル アプリのテストでは、[Appium Java-client API](#) の IOSDriver クラスで利用可能なメソッドを Silk Test Workbench はサポートします。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- プリミティブ型 (boolean、integer、long、double、string)

プリミティブ型 (int など) とオブジェクト タイプ (java.lang.Integer など) の両方がサポートされます。プリミティブ型は必要に応じて拡大変換されます。たとえば、int が必要な場所で long を渡すことができます。
- 列挙型

列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが列挙型 ScreenOrientation のパラメータを必要とする場合、次の文字列値を使用できます：LANDSCAPE、PORTRAIT。
- リスト

リスト、配列、または可変長引数のパラメータを持つメソッドを呼び出すことができます。リストの要素がターゲットの配列型に代入可能な場合、配列型への変換は自動的に行われます。

戻り値

戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

例

次のサンプル コードには、動的呼び出しを使用する共通の例を示します。

```
' VB .NET code
' Getting the page source
```

```
Dim pageSource As String = _desktop.MobileDevice("//
MobileDevice").Invoke("getPageSource")

' Resetting an app
_desktop.MobileDevice("//MobileDevice").Invoke("resetApp")

' Changing the device orientation
_desktop.MobileDevice("//MobileDevice").Invoke("rotate", "LANDSCAPE")
_desktop.MobileDevice("//MobileDevice").Invoke("rotate", "PORTRAIT")

' Dynamic invoke on MobileObject (calls get redirected to the underlying
web element for WebDriver)
With _desktop.MobileDevice("//MobileDevice")
    .MobileObject("CheckBox 2").Invoke("click")
End With
```

モバイル Web サイトでのオブジェクトのクリック

自動テストの記録と再生中にオブジェクトをクリックするとき、モバイル Web サイトではデスクトップ Web サイトと比較して、次のような困難があります。

- 拡大/縮小率やデバイス ピクセル比が異なる
- さまざまなモバイル デバイスによって画面サイズが異なる
- モバイル デバイス間でのフォントとグラフィックサイズが異なる (通常、デスクトップ ブラウザの Web サイトよりも小さい)。
- さまざまなモバイルデバイスによってピクセル サイズと解像度が異なる

Silk Test Workbench は、このような困難をものともせず、モバイル Web サイトの適切なオブジェクトをクリックできます。

モバイル デバイスでテストを記録するときに、Silk Test Workbench は Click の記録時に座標を記録しません。ただし、クロス ブラウザ テストの場合、再生中に座標が許されています。また、Click に座標を手動で追加することもできます。Silk Test Workbench は、これらの座標をオブジェクトの HTML 座標として解釈します。モバイル デバイスのテストの再生時に BrowserWindow の内側の適切なオブジェクトをクリックするために、Silk Test Workbench はオブジェクトの HTML 座標に現在の拡大/縮小率を適用します。デバイスのピクセル座標は、オブジェクトの HTML 座標に現在の拡大/縮小率をかけた座標です。

モバイル Web サイトの現在表示されている領域にオブジェクトが表示されていない場合、Silk Test Workbench は Web サイトの適切な位置にスクロールします。

例

HTML ページで 100 x 20 ピクセルの固定サイズの DomButton をテストするコードを以下に示します。

VB

```
DomButton domButton = _desktop.Find(Of DomButton)("locator for the
button");
domButton.Click(MouseButton.LEFT, new Point(50, 10));
```

異なるモバイル デバイスまたは異なる拡大/縮小率で再生すると、たとえば DomButton は、デバイス画面上では実際は 10 ピクセルの幅かもしれませんが、現在の拡大/縮小率の影響を受けず、上記のコードを使用したときに Silk Test Workbench は要素の中央をクリックします。これは、Silk Test Workbench が座標を HTML 座標として解釈し、現在の拡大/縮小率を適用するためです。

既存のモバイル Web テストの使用方法

Silk Test 17.0 以降では、モバイル Web テストの扱いが、前のバージョンの Silk Test とは異なります。この変更により、以前のモバイル Web テストが Silk Test 17.0 以降では動作しなくなる可能性があります。このトピックでは、Silk Test 17.0 で行われた変更について説明し、既存のモバイル Web テストを Silk Test 17.0 以降で使用できるように変更する方法を説明します。

Silk Test 17.0 でモバイル Web テストに対して行われた変更は、以下の通りです。

- Silk Test の以前のバージョンでは、Windows マシンに USB で接続された iOS デバイスをテストすることができました。Silk Test 17.0 以降では、OS X マシン (Mac) に接続された iOS デバイスに対してのみテストすることができます。
- 以前のバージョンの Silk Test を使用して Android デバイス上のモバイル Web アプリケーションをテストしていた場合、Silk Test 17.0 以降で Web アプリケーションをテストするには、Android デバイスのプロキシ設定を手動で削除する必要があります。Silk Test 17.0 以降では、プロキシは使用しません。プロキシが設定されていると、「プロキシ サーバーに接続できません」というメッセージがデバイスに表示されます。

.NET のサポート

Silk Test は、以下の .NET アプリケーションのテストを組み込みでサポートしています。

- Windows Forms (Win Forms) アプリケーション
- Windows Presentation Foundation (WPF) アプリケーション
- Microsoft Silverlight アプリケーション

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

Windows Forms のサポート

Silk Test Workbench は、.NET スタンドアロン アプリケーションとノートタッチ Windows Forms (Win Forms) アプリケーションのテストを組み込みでサポートしています。ただし、スタンドアロン アプリケーションでは、side-by-side 実行はサポートされていません。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

オブジェクト解決

アプリケーション中の要素に指定された name が使用可能な場合、ロケーターの automationId 属性として使用されます。この結果、多くのオブジェクトは、この属性のみを使用して一意に識別できます。

サポートするコントロール

Win Forms テストで使用可能な記録/再生コントロールの完全な一覧については、「[Windows Forms クラス リファレンス](#)」を参照してください。

Windows Forms クラス リファレンス


Windows Forms アプリケーションを設定すると、Silk Test Workbench は標準の Windows Forms コントロールのテストのサポートを組み込みで提供します。

Windows Forms アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Windows Forms アプリケーションがサポートする属性は次のとおりです。

- automationid
- caption
- windowid
- priorlabel (caption のないコントロールの場合、自動的に priorlabel が caption として使用されます。caption のあるコントロールの場合、caption を使う方が簡単な場合があります。)

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

Windows Forms アプリケーションのカスタム属性


Windows Forms アプリケーションは、あらかじめ定義された自動化用プロパティ automationId を使用して、Windows Forms コントロールに対して安定した識別子を指定します。

Silk Test Workbench は、ロケーターを識別するために、自動的にこのプロパティを使用します。Windows Forms アプリケーションのロケーターは次のようになります。

```
/FormsWindow//PushButton[@automationId='btnBasicControls']
```

Windows Forms メソッドの動的な呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある SetTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

Invoke メソッド

Windows Forms または WPF コントロールでは、Invoke メソッドを使用して、以下のメソッドを呼び出すことができます。

- MSDN が定義するコントロールのパブリック メソッド。
- MSDN が定義する静的パブリック メソッド。
- ユーザーが定義する任意の型の静的パブリック メソッド。

Invoke メソッドの最初の例

Silk Test Workbench の DataGrid 型のオブジェクトでは、MSDN が System.Windows.Forms.DataGrid 型に定義しているすべてのメソッドを呼び出すことができます。

System.Windows.Forms.DataGrid クラスのメソッド IsExpanded を呼び出すには、次のコードを使用します。

```
//VB .NET code
Dim isExpanded As Boolean = dataGrid.Invoke("IsExpanded", 3)
```

Invoke メソッドの 2 番目の例

AUT 内の静的メソッド String.Compare(String s1, String s2) を呼び出すには、次のコードを使用します。

```
//VB .NET code
Dim result as Integer = (Integer)
mainWindow.Invoke("System.String.Compare", "a", "b")
```

Invoke メソッドの 3 番目の例

この例では、ユーザーが生成したメソッド GetContents を動的に呼び出す方法を示します。

テスト対象アプリケーション (AUT) のコントロールの操作に使用するコードを作成できます (この例では UltraGrid)。UltraGrid の内容を取得するために、複雑な動的呼び出しを作成するのではなく、新しいメソッド GetContents を生成し、この新しいメソッドを動的に呼び出すことができます。

Visual Studio で、AUT 内の次のコードによって GetContents メソッドを UltraGridUtil クラスのメソッドとして定義します。

```
//C# code, because this is code in the AUT
namespace UltraGridExtensions {
    public class UltraGridUtil {
        /// <summary>
        /// Retrieves the contents of an UltraGrid as nested list
        /// </summary>
        /// <param name="grid"></param>
        /// <returns></returns>
        public static List<List<string>>
GetContents(Infragistics.Win.UltraWinGrid.UltraGrid grid) {
            var result = new List<List<string>>();
            foreach (var row in grid.Rows) {
                var rowContent = new List<string>();
                foreach (var cell in row.Cells) {
                    rowContent.Add(cell.Text);
                }
                result.Add(rowContent);
            }
        }
    }
}
```

```
        return result;
    }
}
}
```

UltraGridUtil クラスのコードを AUT に追加する必要があります。これは、次のようにして行います。

- アプリケーション開発者は、クラスのコードを AUT にコンパイルできます。アセンブリがすでにロードされている必要があります。
- テストの実行時に AUT にロードされる新しいアセンブリを作成できます。

アセンブリをロードするには、次のコードを使用します。

```
FormsWindow.LoadAssembly(String assemblyFileName)
```

次のようにして、フルパスで指定してアセンブリをロードします。

```
mainWindow.LoadAssembly("C:/temp/ultraGridExtensions.dll")
```

Location メソッドを使用してアセンブリの場所を見つけることもできます。

```
//VB.NET code
Dim assemblyLocation =
GetType(UltraGridExtensions.UltraGridUtil).Assembly.Location
mainWindow.LoadAssembly(assemblyLocation)
```

UltraGridUtil クラスのコードが AUT 内にある場合は、次のコードをテストスクリプトに追加して、GetContents メソッドを呼び出すことができます。

```
var contents = (IList)
mainWindow.Invoke("UltraGridExtensions.UltraGridUtil.GetContents",
ultraGrid);
```

Invoke メソッドを呼び出す mainWindow オブジェクトは、AUT を特定しているだけなので、同じ AUT の他のオブジェクトに置き換えてもかまいません。

InvokeMethods メソッド

Windows Forms または WPF コントロールでは、InvokeMethods メソッドを使用して、ネストされたメソッドのシーケンスを呼び出すことができます。以下のメソッドを呼び出すことができます。

- MSDN が定義するコントロールのパブリック メソッド。
- MSDN が定義する静的パブリック メソッド。
- ユーザーが定義する任意の型の静的パブリック メソッド。

例：カスタム データ グリッドのセルの内容のテキストでの取得

Infragistics ライブラリのカスタム データ グリッドのセルの内容をテキストで取得するには、AUT で次の C# コードを使用できます。

```
string cellText = dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
```

次の C# コードのサンプルは、最初の行の 3 番目のセルの内容をテキストで取得します。

```
string cellText = dataGrid.Rows[0].Cells[2];
```

InvokeMethods メソッドを使用して同じ例をスクリプト化すると、比較的複雑なスクリプトになります。これは、対応するパラメータを持つ 5 つのメソッドを InvokeMethods メソッドに渡さなければならないためです。

```
' VB .NET code
Dim dataGrid = mainWindow.WPFControl("@automationId='Custom Data Grid'")
```

```

' Get text contents of third cell in first row.
Dim rowIndex = 0
Dim column = 2

Dim methodNames = New List(Of String)()
methodNames.Add("Rows") ' Get the list of rows
from the grid.
methodNames.Add("get_Item") ' Get a specific row from
the list of rows by using the indexer method.
methodNames.Add("Cells") ' Get the list of cells from
the the row.
methodNames.Add("get_Item") ' Get a specific cell from
the list of cells by using the indexer method.
methodNames.Add("Text") ' Get the text of the cell.

Dim parameters = New List(Of List(Of Object))()
parameters.Add(New List(Of Object)()) ' Parameters
for the Rows property.
parameters.Add(New List(Of Object) From {rowIndex}) ' Parameters
for the get_Item method.
parameters.Add(New List(Of Object)()) ' Parameters
for the Cells property.
parameters.Add(New List(Of Object) From {columnIndex}) ' Parameters
for the get_Item method.
parameters.Add(New List(Of Object)()) ' Parameters
for the Text property.

Dim cellText As String = dataGrid.InvokeMethods(methodNames,
parameters)

```

このような場合に、より簡単にするアプローチは、テスト対象アプリケーションにコードを追加して、InvokeMethods メソッドを使用することです。たとえば、GetCellText メソッドを AUT に追加します。

```
// C# code, if the AUT is implemented in C#.
public static string GetCellText(Infragistics.Win.UltraWinGrid.UltraGrid
dataGrid, int rowIndex, int columnIndex) {
    return dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
}

```

```
' VB code, if the AUT is implemented in VB.
public static string GetCellText(Infragistics.Win.UltraWinGrid.UltraGrid
dataGrid, int rowIndex, int columnIndex) {
    return dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
}

```

テスト スクリプトから GetCellText メソッドを動的に呼び出して、セルの内容をテキストで取得します。

```
'VB .NET code
Dim cellText As String = mainWindow.Invoke("GetCellText", dataGrid,
rowIndex, columnIndex)

```

詳細については、「テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする」を参照してください。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。

- MSDN が定義するコントロールのパブリック メソッドとプロパティ。
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型
 - Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point や Rect など) が含まれます。
- 列挙型
 - 列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが .NET 列挙型 System.Windows.Visibility のパラメータを必要とする場合、次の文字列値を使用できます： Visible、Hidden、Collapsed。
- .NET 構造体とオブジェクト
 - .NET 構造体とオブジェクトパラメータはリストとして渡す必要があります。リスト内の要素は、テストアプリケーションの .NET オブジェクトで定義されているコンストラクタの 1 つと一致しなければなりません。たとえば、メソッドが .NET 型 System.Windows.Vector のパラメータを必要とする場合、2 つの整数値を持つリストを渡すことができます。これが機能するのは、System.Windows.Vector 型が 2 つの整数値を引数に取るコンストラクタを持つためです。
- その他のコントロール
 - コントロールパラメーターは、TestObject として渡したり、返したりできます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

Windows Presentation Foundation (WPF) のサポート

Silk Test Workbench は、Windows Presentation Foundation (WPF) アプリケーションのテストを組み込みでサポートしています。Silk Test Workbench は、スタンドアロン WPF アプリケーションをサポートしており、.NET バージョン 3.5 以降に組み込まれているコントロールを記録し、再生できます。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

サポートするコントロール

WPF テストで使用可能なコントロールの完全な一覧については、「[WPF クラス リファレンス](#)」を参照してください。

Silk Test Workbench WPF がサポートするすべての WPF クラスは、WPFWindow や WPFListBox のように接頭辞 *WPF* で始まります。

WPF コントロールでサポートされるメソッドとプロパティは、実際の実装とランタイム状態によって異なります。メソッドとプロパティは、対応するクラスに対して定義されたリストと異なる場合があります。特定の状況でサポートされるメソッドとプロパティを判別するには、以下のコードを使用します。

- `GetPropertyList()`

- `GetDynamicMethodList()`

WPF の詳細については、[MSDN](#) を参照してください。

WPF クラス リファレンス

WPF アプリケーションを設定すると、Silk Test Workbench は標準の WPF コントロールのテストのサポートを組み込みで提供します。

Windows Presentation Foundation (WPF) アプリケーションの属性

WPF アプリケーションがサポートする属性は次のとおりです。

- *automationId*
- *caption*
- *className*
- *name*
- すべての動的ロケータ属性。



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード `?` および `*` をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

オブジェクト解決

WPF スクリプト内のコンポーネントを識別するために、*automationId*、*caption*、*className*、あるいは *name* を指定できます。アプリケーション中の要素に指定された *name* が利用可能な場合、ロケータの *automationId* 属性として使用されます。この結果、多くのオブジェクトは、この属性のみを使用して一意に識別できます。たとえば、*automationId* を持つロケータは、以下のようになります：`//`

```
WPFButton[@automationId='okButton']"
```

automationId や他の属性を定義した場合、再生中に *automationId* だけが使用されます。 *automationId* が定義されていない場合には、コンポーネントを解決するのに *name* が使用されます。 *name* も *automationId* もどちらも定義されていない場合には、*caption* 値が使用されます。 *caption* が定義されていない場合は、*className* が使用されます。 *automationId* は非常に役立つプロパティであるため、使用することを推奨します。

属性の種類	説明	例
<i>automationId</i>	テスト アプリケーションの開発者によって提供された ID	<code>//WPFButton[@automationId='okButton']"</code>
<i>name</i>	コントロールの名前。Visual Studio デザイナは、デザイナ上で作成されたすべてのコントロールに自動的に名前を割り当てます。アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この名前を使用します。	<code>//WPFButton[@name='okButton']"</code>
<i>caption</i>	コントロールが表示するテキスト。複数の言語にロー	<code>//WPFButton[@automationId='Ok']"</code>

属性の種類	説明	例
	<p>カライズされたアプリケーションをテストする場合、caption の代わりに automationId や name 属性を使用することを推奨します。</p>	
className	<p>WPF の .NET 単純クラス名 (名前空間なし)。クラス名属性を使用すると、Silk Test Workbench が解決する標準 WPF コントロールから派生したカスタム コントロールを識別するのに役立ちます。</p>	<pre>//WPFButton[@className='MyCustomButton']"</pre>

Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性をこの表に示した順番に使用して WPF コントロールのロケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、Silk Test Workbench がロケータを作成する際には *automationId* が使用されません。

以下の例では、アプリケーション開発者がアプリケーションの WPF ボタンに対して *name* と *automationId* を XAML コードに定義する方法を示します。

```
<Button Name="okButton" AutomationProperties.AutomationId="okButton"
Click="okButton_Click">Ok</Button>
```

WPF アプリケーションのカスタム属性

WPF アプリケーションは、あらかじめ定義された自動化用プロパティ AutomationProperties.AutomationId を使用して、次のように WPF コントロールに対して安定した識別子を指定します。

```
<Window x:Class="Test.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="MainWindow" Height="350" Width="525">
<Grid>
<Button AutomationProperties.AutomationId="AID_buttonA">The
Button</Button>
</Grid>
</Window>
```

Silk Test Workbench は、ロケータを識別するために、自動的にこのプロパティを使用します。WPF アプリケーションのロケータは次のようになります。

```
/WPFWindow[@caption='MainWindow']/WPFButton[@automationId='AID_buttonA']
```

WPFItemsControl クラスから派生したクラス

Silk Test Workbench は、2 つの方法を使用して WPFItemsControl から派生したクラス (WPFListBox、WPFTreeView、WPFMenu など) を操作することができます。

- コントロールでの作業

ほとんどのコントロールには、標準的なユースケースのためのメソッドやプロパティがあります。項目は、テキストや索引によって識別されます。

例：

```
listBox.Select("Banana")
listBox.Select(2)
```

```
tree.Expand("/Fruit/Banana")
```

- WPFListBoxItem、WPFTreeViewItem、WPFMenuItem などの個々の項目での作業

高度なユースケースの場合、個々の項目を使用します。たとえば、リスト ボックスの特定の項目のコンテキスト メニューを開いたり、項目に相対的な場所をクリックしたりする場合に個々の項目を使用します。

例：

```
Dim item as WPFListBoxItem = listBox.WPFListBoxItem("banana")
item.OpenContextMenu()
```

カスタム WPF コントロール

一般的に、Silk Test Workbench では、すべての標準 WPF コントロールの記録と再生がサポートされています。

Silk Test Workbench は、カスタム コントロールが実装された方法を基にしてカスタム コントロールを処理します。次の方法を使用してカスタム コントロールを実装することができます。

- UserControl から派生したクラスを定義する

複合コントロールを作成する典型的な方法です。Silk Test Workbench は、これらのユーザー コントロールを WPFUserControl として認識し、含まれるコントロールを完全にサポートしています。

- ListBox などの標準 WPF コントロールから派生したクラスを定義する

Silk Test Workbench は、これらのコントロールを派生元の標準 WPF コントロールのインスタンスとして扱います。ユーザー コントロールの振る舞いがその基底クラスの実装と大きく異なる場合には、子の記録、再生、解決は機能しない可能性があります。

- テンプレートを使用して視覚デザインを変更した標準コントロールを使用する

低レベルの再生が機能しない可能性があります。その場合には、「高レベル」再生モードに切り替えます。再生モードを変更するには、**オプション** ダイアログ ボックスを使用して、**再生全般** 設定リストで **再生モード** を変更します。

Silk Test Workbench は、一般的に機能テストに無関係なコントロールは除外します。たとえば、レイアウトを目的として使用されるコントロールは含まれません。しかし、カスタム コントロールが除外されたクラスから派生している場合、除外されたコントロールを記録/再生の対象とするためには、関連する WPF クラスの名前を指定します。

WPF メソッドの動的な呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。



注： 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。

オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には `GetProperty` メソッドを、動的プロパティの設定には `SetProperty` メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、`GetPropertyList` メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを `String` 型の入力パラメータとして設定する必要がある `SetTitle` というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```



注： 通常、ほとんどのプロパティは読み取り専用で、設定できません。



注： ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

Invoke メソッド

Windows Forms または WPF コントロールでは、Invoke メソッドを使用して、以下のメソッドを呼び出すことができます。

- MSDN が定義するコントロールのパブリック メソッド。
- MSDN が定義する静的パブリック メソッド。
- ユーザーが定義する任意の型の静的パブリック メソッド。

Invoke メソッドの最初の例

Silk Test Workbench の `DataGrid` 型のオブジェクトでは、MSDN が `System.Windows.Forms.DataGrid` 型に定義しているすべてのメソッドを呼び出すことができます。

`System.Windows.Forms.DataGrid` クラスのメソッド `IsExpanded` を呼び出すには、次のコードを使用します。

```
//VB .NET code  
Dim isExpanded As Boolean = dataGrid.Invoke("IsExpanded", 3)
```

Invoke メソッドの 2 番目の例

AUT 内の静的メソッド `String.Compare(String s1, String s2)` を呼び出すには、次のコードを使用します。

```
//VB .NET code  
Dim result as Integer = (Integer)  
mainWindow.Invoke("System.String.Compare", "a", "b")
```

Invoke メソッドの 3 番目の例

この例では、ユーザーが生成したメソッド `GetContents` を動的に呼び出す方法を示します。

テスト対象アプリケーション (AUT) のコントロールの操作に使用するコードを作成できます (この例では `UltraGrid`)。 `UltraGrid` の内容を取得するために、複雑な動的呼び出しを作成するのではなく、新しいメソッド `GetContents` を生成し、この新しいメソッドを動的に呼び出すことができます。

Visual Studio で、AUT 内の次のコードによって `GetContents` メソッドを `UltraGridUtil` クラスのメソッドとして定義します。

```
//C# code, because this is code in the AUT  
namespace UltraGridExtensions {  
    public class UltraGridUtil {
```

```

/// <summary>
/// Retrieves the contents of an UltraGrid as nested list
/// </summary>
/// <param name="grid"></param>
/// <returns></returns>
public static List<List<string>>
GetContents(Infragistics.Win.UltraWinGrid.UltraGrid grid) {
    var result = new List<List<string>>();
    foreach (var row in grid.Rows) {
        var rowContent = new List<string>();
        foreach (var cell in row.Cells) {
            rowContent.Add(cell.Text);
        }
        result.Add(rowContent);
    }
    return result;
}
}
}
}

```

UltraGridUtil クラスのコードを AUT に追加する必要があります。これは、次のようにして行います。

- アプリケーション開発者は、クラスのコードを AUT にコンパイルできます。アセンブリがすでにロードされている必要があります。
- テストの実行時に AUT にロードされる新しいアセンブリを作成できます。

アセンブリをロードするには、次のコードを使用します。

```
FormsWindow.LoadAssembly(String assemblyFileName)
```

次のようにして、フルパスで指定してアセンブリをロードします。

```
mainWindow.LoadAssembly("C:/temp/ultraGridExtensions.dll")
```

Location メソッドを使用してアセンブリの場所を見つけることもできます。

```
//VB.NET code
Dim assemblyLocation =
GetType(UltraGridExtensions.UltraGridUtil).Assembly.Location
mainWindow.LoadAssembly(assemblyLocation)
```

UltraGridUtil クラスのコードが AUT 内にある場合は、次のコードをテスト スクリプトに追加して、GetContents メソッドを呼び出すことができます。

```
var contents = (IList)
mainWindow.Invoke("UltraGridExtensions.UltraGridUtil.GetContents",
ultraGrid);
```

Invoke メソッドを呼び出す mainWindow オブジェクトは、AUT を特定しているだけなので、同じ AUT の他のオブジェクトに置き換えてもかまいません。

InvokeMethods メソッド

Windows Forms または WPF コントロールでは、InvokeMethods メソッドを使用して、ネストされたメソッドのシーケンスを呼び出すことができます。以下のメソッドを呼び出すことができます。

- MSDN が定義するコントロールのパブリック メソッド。
- MSDN が定義する静的パブリック メソッド。
- ユーザーが定義する任意の型の静的パブリック メソッド。

例 : カスタム データ グリッドのセルの内容のテキストでの取得

Infragistics ライブラリのカスタム データ グリッドのセルの内容をテキストで取得するには、AUT で次の C# コードを使用できます。

```
string cellText = dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
```

次の C# コードのサンプルは、最初の行の 3 番目のセルの内容をテキストで取得します。

```
string cellText = dataGrid.Rows[0].Cells[2];
```

InvokeMethods メソッドを使用して同じ例をスクリプト化すると、比較的複雑なスクリプトになります。これは、対応するパラメータを持つ 5 つのメソッドを InvokeMethods メソッドに渡さなければならないためです。

```
' VB .NET code
Dim dataGrid = mainWindow.WPFControl("@automationId='Custom Data Grid'")

' Get text contents of third cell in first row.
Dim rowIndex = 0
Dim column = 2

Dim methodNames = New List(Of String)()
methodNames.Add("Rows")           ' Get the list of rows
from the grid.
methodNames.Add("get_Item")       ' Get a specific row from
the list of rows by using the indexer method.
methodNames.Add("Cells")         ' Get the list of cells from
the the row.
methodNames.Add("get_Item")       ' Get a specific cell from
the list of cells by using the indexer method.
methodNames.Add("Text")          ' Get the text of the cell.

Dim parameters = New List(Of List(Of Object))()
parameters.Add(New List(Of Object)()) ' Parameters
for the Rows property.
parameters.Add(New List(Of Object) From {rowIndex}) ' Parameters
for the get_Item method.
parameters.Add(New List(Of Object)()) ' Parameters
for the Cells property.
parameters.Add(New List(Of Object) From {columnIndex}) ' Parameters
for the get_Item method.
parameters.Add(New List(Of Object)()) ' Parameters
for the Text property.

Dim cellText As String = dataGrid.InvokeMethods(methodNames,
parameters)
```

このような場合に、より簡単にするアプローチは、テスト対象アプリケーションにコードを追加して、InvokeMethods メソッドを使用することです。たとえば、GetCellText メソッドを AUT に追加します。

```
// C# code, if the AUT is implemented in C#.
public static string GetCellText(Infragistics.Win.UltraWinGrid.UltraGrid
dataGrid, int rowIndex, int columnIndex) {
    return dataGrid.Rows[rowIndex].Cells[columnIndex].Text;
```

```
' VB code, if the AUT is implemented in VB.
public static string GetCellText(Infragistics.Win.UltraWinGrid.UltraGrid
```

```
dataGrid, int rowIndex, int columnIndex) {  
    return dataGrid.Rows[rowIndex].Cells[columnIndex].Text;  
}
```

テスト スクリプトから GetCellText メソッドを動的に呼び出して、セルの内容をテキストで取得します。

```
'VB .NET code  
Dim cellText As String = mainWindow.Invoke("GetCellText", dataGrid,  
rowIndex, columnIndex)
```

詳細については、「テスト対象アプリケーションにコードを追加してカスタム コントロールをテストする」を参照してください。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- MSDN が定義するコントロールのパブリック メソッドとプロパティ。
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型

Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point や Rect など) が含まれます。

- 列挙型

列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが .NET 列挙型 System.Windows.Visibility のパラメータを必要とする場合、次の文字列値を使用できます： Visible、Hidden、Collapsed。

- .NET 構造体とオブジェクト

.NET 構造体とオブジェクト パラメータはリストとして渡す必要があります。リスト内の要素は、テスト アプリケーションの .NET オブジェクトで定義されているコンストラクタの 1 つと一致しなければなりません。たとえば、メソッドが .NET 型 System.Windows.Vector のパラメータを必要とする場合、2 つの整数値を持つリストを渡すことができます。これが機能するのは、System.Windows.Vector 型が 2 つの整数値を引数に取るコンストラクタを持つためです。

- WPF コントロール

WPF コントロール パラメータは TestObject として渡すことができます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。
- すべてのその他の型の場合は文字列

返された .NET オブジェクトに対して ToString を呼び出せば、文字列表現を取得できます。

例

たとえば、アプリケーション開発者が次のメソッドとプロパティを持つ Calculator カスタム コントロールを作成したとします。

```
public void Reset()
public int Add(int number1, int number2)
public System.Windows.Vector StrechVector(System.Windows.Vector vector,
double
factor)
public String Description { get;}
```

テスト担当者は、テスト内からメソッドを直接呼び出すことができます。例：

```
customControl.Invoke("Reset")
Dim sum as Integer = customControl.Invoke("Add", 1, 2)
' the vector can be passed as list of integer
Dim vector = New List(Of Integer)
vector.Add(3)
vector.Add(4)
' returns "6;8" because this is the string representation of the .NET
object
Dim stretchedVector As String = customControl.Invoke("StrechVector",
vector, 2.0)
Dim description As String = customControl.GetProperty("Description")
```

記録/再生の対象とする WPF クラスの設定

Silk Test Workbench は、一般的に機能テストに無関係なコントロールは除外します。たとえば、レイアウトを目的として使用されるコントロールは含まれません。しかし、カスタム コントロールが除外されたクラスから派生している場合、除外されたコントロールを記録/再生の対象とするためには、関連する WPF クラスの名前を指定します。

記録や再生の対象にしたい WPF クラスの名前を指定します。たとえば、*MyGrid* というカスタム クラスが WPF Grid クラスから継承された場合、*MyGrid* カスタム クラスのオブジェクトは記録や再生に使用できません。Grid クラスはレイアウト目的のためにのみ存在し、機能テストとは無関係であるため、Grid オブジェクトは記録や再生に使用できません。この結果、Grid オブジェクトはデフォルトでは公開されません。機能テストに無関係なクラスに基づいたカスタム クラスを使用するには、カスタム クラス (この場合は *MyGrid*) を **OPT_WPF_CUSTOM_CLASSES** オプションに追加します。これによって、記録、再生、検索、プロパティの検証など、すべてのサポートされる操作を指定したクラスに対して実行できるようになります。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。
3. **WPF** をクリックします。
4. **カスタム WPF クラス名** グリッドで、記録や再生中に公開するクラスの名前を入力します。
複数のクラス名を指定する場合にはカンマで区切ります。
5. **OK** をクリックします。

記録/再生時の事前読み込みの設定

記録および再生中に、WPFComboBox や WPFListBox のような WPFItemsControl 内の項目を事前に入力するかどうかを定義します。WPF 自体が特定のコントロールの項目を遅延読み込みするため、項目がビューにスクロールされない場合、それらの項目は Silk Test Workbench では使用できません。ビューにスクロールされないアクセスできない項目にアクセスするには、事前入力をオンにします。これはデフォルトの設定です。ただし、一部のアプリケーションでは Silk Test Workbench によってバックグラウンドで項目が事前入力されると問題が発生し、そのためアプリケーションがクラッシュすることがあります。この場合、事前入力をオフにします。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの 記録 の隣にあるプラス記号 (+) をクリックします。
3. WPF をクリックします。
4. 項目の事前読み込み を はい に設定します。
5. OK をクリックします。

Silverlight アプリケーションのサポート

Microsoft Silverlight (Silverlight) は、リッチ インターネット アプリケーションを記述し、実行するためのアプリケーション フレームワークで、Adobe Flash と同様の機能と目的を備えています。Silverlight の実行時環境は、大部分の Web ブラウザでプラグインとして使用できます。

Silk Test Workbench は、Silverlight アプリケーションのテストを組み込みでサポートしています。Silk Test Workbench は、ブラウザ内部と同様ブラウザ外部でも実行される Silverlight アプリケーションをサポートしており、.NET バージョン 3.5 以降でコントロールを記録し、再生できます。

Silverlight をベースとする以下のアプリケーションがサポートされます。


- Internet Explorer で実行される Silverlight アプリケーション
- ブラウザー外実行 Silverlight アプリケーション

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

サポートするコントロール


Silk Test Workbench は、Silverlight コントロールの記録と再生をサポートしています。

Silverlight テストで使用可能なコントロールの完全な一覧については、「[Silverlight クラス リファレンス](#)」を参照してください。

 **注:** Silk Test 14.0 以降では、Silk Test Workbench は、画面上で操作可能でかつ表示されている Silverlight コントロールのみを認識します。この変更は、Silk Test 14.0 より前のバージョンの Silk Test を使用して記録されたテストの動作に影響を与える可能性があります。Silk Test 14.0 以降を使用してこのようなテストを実行するには、不可視な、または利用可能でないすべての Silverlight コントロールをテストから削除してください。

前提条件

Microsoft Windows XP で Silverlight アプリケーションをテストする場合、サービスパック 3 をインストールし、Windows 7 で提供される Microsoft User Interface Automation の Windows XP 用の更新プログラムを適用する必要があります。更新プログラムは、<http://www.microsoft.com/download/en/details.aspx?id=13821> からダウンロードできます。

 **注:** Microsoft User Interface Automation は、Silverlight サポート用にインストールする必要があります。Windows オペレーティング システムを使用していて、Silverlight サポートが機能しない場合は、使用しているオペレーティング システム用の Microsoft User Interface Automation の更新プログラムを <http://support.microsoft.com/kb/971513> からダウンロードしてインストールしてください。

Silverlight クラス リファレンス

Silverlight アプリケーションを設定すると、Silk Test Workbench は標準の Silverlight コントロールのテストのサポートを組み込みで提供します。

Silverlight コントロールを識別するためのロケータ属性

Silverlight コントロールでサポートされているロケータ属性は次のとおりです。

- *automationId*
- *caption*
- *className*
- *name*
- すべての動的ロケータ属性



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

Silverlight スクリプト内のコンポーネントを識別するために、*automationId*、*caption*、*className*、*name*、または任意の動的ロケータ属性を指定できます。*automationId* はアプリケーション開発者が設定します。たとえば、*automationId* を持つロケータは、以下のようになります：
`// SLButton[@automationId="okButton"]`

automationId は一般に非常に有用で安定した属性であるため、使用することを推奨します。

属性の種類	説明	例
<i>automationId</i>	テスト対象アプリケーションの開発者によって設定される識別子。Visual Studio デザイナは、デザイナー上で作成されたすべてのコントロールに自動的に <i>automationId</i> を割り当てます。アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この ID を使用します。	<code>// SLButton[@automationId="okButton"]</code>
<i>caption</i>	コントロールが表示するテキスト。複数の言語にローカライズされたアプリケーションをテストする場合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	<code>//SLButton[@caption="Ok"]</code>
<i>className</i>	Silverlight コントロールの .NET 単純クラス名 (名前空間なし)。 <i>className</i> 属性を使用すると、Silk Test Workbench が解決する標準 Silverlight コントロールから派生したカスタム コントロールを識別するのに役立ちます。	<code>// SLButton[@className='MyCustomButton']</code>
<i>name</i>	コントロールの名前。テスト対象アプリケーションの開発者によって設定されます。	<code>//SLButton[@name="okButton"]</code>



注目: XAML コードの *name* 属性は、ロケータ属性 *name* ではなく、ロケータ属性 *automationId* にマップされます。

Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性をこの表に示した順番に使用して Silverlight コントロールのロケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、*automationId* が固有の場合は Silk Test Workbench がロケータを作成する際に使用されます。


以下の表は、アプリケーション開発者がテキスト「Ok」を持つ Silverlight ボタンをアプリケーションの XAML コードに定義する方法を示しています。

オブジェクトの XAML コード	Silk Test からオブジェクトを検索するためのロケータ
<code><Button>Ok</Button></code>	<code>//SLButton[@caption="Ok"]</code>
<code><Button Name="okButton">Ok</Button></code>	<code>//SLButton[@automationId="okButton"]</code>
<code><Button AutomationProperties.AutomationId="okButton">Ok</Button></code>	<code>//SLButton[@automationId="okButton"]</code>

オブジェクトの XAML コード	Silk Test からオブジェクトを検索するためのロケータ
<Button AutomationProperties.Name="okButton">Ok< /Button>	//SLButton[@name="okButton"]

Silverlight メソッドの動的呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある SetTitle というメソッドを呼び出すには、次のように入力します：

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型。

Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point、Rect など) が含まれます。

- 列挙型。

列挙パラメータは文字列として渡す必要があります。文字列は、列挙値の名前と一致しなければなりません。たとえば、メソッドが .NET 列挙型 System.Windows.Visibility のパラメータを必要とする場合には、Visible、Hidden、Collapsed の文字列値を使用できます。

- .NET 構造体とオブジェクト。

.NET 構造体とオブジェクトパラメータはリストとして渡します。リスト内の要素は、テスト アプリケーションの .NET オブジェクトで定義されているコンストラクタの 1 つと一致しなければなりません。たとえば、メソッドが .NET 型 System.Windows.Vector のパラメータを必要とする場合、2 つの整数値を持つリストを渡すことができます。これが機能するのは、System.Windows.Vector 型が 2 つの整数値を引数に取るコンストラクタを持つためです。

- その他のコントロール。

コントロールパラメータは TestObject として渡すことができます。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- MSDN が定義する AutomationElement クラスのすべてのパブリック メソッドとプロパティ。詳細については、<http://msdn.microsoft.com/en-us/library/system.windows.automation.automationelement.aspx> を参照してください。
- MSUIA が公開するすべてのメソッドとプロパティ。利用可能なメソッドとプロパティは「パターン」で分類されます。パターンとは、MSUIA 固有の用語です。すべてのコントロールは、いくつかのパターンを実装します。一般的なパターンについての概要およびすべての利用可能なパターンについては、<http://msdn.microsoft.com/en-us/library/ms752362.aspx> を参照してください。カスタム コントロールの開発者は、MSUIA パターン セットを実装することによって、カスタム コントロールのテスト サポートを提供できます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。
- すべてのその他の型の場合は文字列。

この文字列表現を取得するには、テスト対象アプリケーションの返された .NET オブジェクトに対して ToString メソッドを呼び出します。

例

Silverlight の TabItem。これは TabControl の項目です。

```
tabItem.Invoke("SelectedItemPattern.Select")  
mySilverlightObject.GetProperty("IsPassword")
```

Silverlight でのスクロール

Silk Test Workbench では、Silverlight コントロールに応じて、2 種類のスクロール方法とプロパティを提供します。

- 1 つめの種類のコントロールには、それ自体でスクロール可能なコントロールが含まれ、スクロールバーは子として明示的に表示されません。たとえば、コンボ ボックス、ペイン、リスト ボックス、ツリー コントロール、データグリッド、オート コンプリット ボックスなどがあります。
- 2 つめの種類のコントロールには、それ自体ではスクロール不可能なコントロールが含まれ、スクロール用にスクロールバーが子として表示されます。たとえば、テキスト フィールドがあります。

Silk Test Workbench にこのような違いがあるのは、Silk Test Workbench のコントロールがこの 2 通りの方法でスクロールを実装するためです。

スクロールをサポートするコントロール

この場合、スクロール方法とプロパティは、スクロールバーを含むコントロールで使用できます。したがって、Silk Test Workbench ではスクロールバー オブジェクトは表示されません。

使用例

以下のコマンドでは、リスト ボックスが一番下までスクロールされます。

```
listBox.SetVerticalScrollPercent(100)
```

以下のコマンドでは、リスト ボックスが 1 ユニットずつ下方へスクロールされます。

```
listBox.ScrollVertical(ScrollAmount.SmallIncrement)
```

スクロールをサポートしないコントロール

この場合、スクロールバーが表示されます。コントロール自体で可能なスクロール方法とプロパティはありません。水平スクロールバーと垂直スクロールバーの各オブジェクトを使用すると、対応する API 関数でパラメータとして増分または減分、または最終位置を指定することでコントロール内をスクロールできます。増分または減分として ScrollAmount 列挙の値を使用できます。詳細については、Silverlight の製品マニュアルを参照してください。最終位置は、オブジェクトの位置に関連し、アプリケーション設計者によって定義されます。

使用例

以下のコマンドでは、テキスト ボックス内の垂直スクロールバーが 15 の位置までスクロールされます。

```
textBox.SLVerticalScrollBar().ScrollToPosition(15)
```

以下のコマンドでは、テキスト ボックス内の垂直スクロールバーが一番下までスクロールされます。

```
textBox.SLVerticalScrollBar().ScrollToMaximum()
```

Silverlight アプリケーションのテスト時のトラブルシューティング

Silk Test Workbench で Silverlight アプリケーションの内部を確認できず、記録時に緑色の矩形領域が描画されない

次の理由により、Silk Test Workbench は Silverlight アプリケーションの内部を確認できなくなっています。

原因	解決策
使用している Silverlight のバージョンが 3 以前である	Silverlight 3 (Silverlight Runtime 4) または Silverlight 4 (Silverlight Runtime 4) を使用します。
使用している Silverlight アプリケーションがウィンドウレスモードで実行されている	Silk Test Workbench は、ウィンドウレスモードで実行される Silverlight アプリケーションをサポートしません。このようなアプリケーションをテストするには、Silverlight アプリケーションが実行されている Web サイトを変更する必要があります。つまり、Silverlight アプリケーションがホストされている HTML または ASPX ファイルのオブジェクト タグの windowless パラメータを false に設定する必要があります。 以下のコードは、windowless パラメータを false に設定する例を示します。 <pre><object ...> <param name="windowless" value="false"/> ... </object></pre>

Visual COBOL のサポート

Silk Test Workbench は、Visual COBOL アプリケーションに対するテストの記録と再生をサポートします。.NET COBOL から Silk Test Workbench API を使用して、次の Visual COBOL アプリケーションに対して自動テストのスクリプトを記述することもできます。

- ダイアログ システム アプリケーション

- CGI アプリケーション
- COBOL Win32 アプリケーション
- .NET COBOL - WPF および Windows Forms アプリケーション
- COBOL JVM - Swing アプリケーション
- バックエンドにある COBOL を呼び出す非 COBOL のフロントエンド アプリケーション



注: コントロールによっては、Silk Test Workbench は低レベルの記録のみをサポートします。

サポートする Visual COBOL のバージョンについての情報は、『[リリースノート](#)』を参照してください。

Rumba のサポート

Rumba は、世界トップクラスの Windows デスクトップ端末エミュレーション ソリューションです。Silk Test は、Rumba の記録および再生を組み込みでサポートしています。

Rumba でのテスト時には、以下の点を考慮してください。

- Rumba のバージョンは、Silk Test のバージョンと互換性がある必要があります。バージョン 8.1 以前の Rumba はサポートされていません。
- Rumba のグリーン スクリーンの周囲にあるコントロールはすべて WPF の基本機能 (または Win32) を使用しています。
- サポートされている Rumba デスクトップ タイプは、以下のとおりです。
 - メインフレーム ディスプレイ
 - AS400 ディスプレイ
 - Unix ディスプレイ

Rumba テストで使用できる記録および再生のコントロールの完全な一覧については、「[Rumba クラス リファレンス](#)」を参照してください。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

Rumba クラス リファレンス

Rumba アプリケーションを設定すると、Silk Test Workbench は標準の Rumba コントロールのテストのサポートを組み込みで提供します。

Rumba の有効化と無効化

Rumba は、世界トップクラスの Windows デスクトップ端末エミュレーション ソリューションです。Rumba は、メインフレーム、ミッドレンジ、UNIX、Linux、および HP サーバーとの接続ソリューションを提供します。

サポートの有効化

Rumba スクリプトを記録および再生する前に、サポートを有効にする必要があります。

1. Rumba デスクトップ クライアント ソフトウェア バージョン 8.1 以降をインストールします。
2. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > Rumba プラグイン > Silk Test Rumba プラグインの有効化**、または (Microsoft Windows 10) **スタート > Silk > Silk Test Rumba プラグインの有効化** をクリックします。


サポートの無効化

(Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > Rumba プラグイン > Silk Test Rumba プラグインの無効化**、または (Microsoft Windows 10) **スタート > Silk > Silk Test Rumba プラグインの無効化** をクリックします。

Rumba コントロールを識別するためのロケータ属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。サポートされている属性は次のとおりです。

caption	コントロールが表示するテキスト。
priorlabel	フォームの入力フィールドには通常入力の目的を説明するラベルがあるため、 priorlabel の目的は隣接するラベル フィールド RumbaLabel のテキストによってテキスト入力フィールド RumbaTextField を識別することです。テキスト フィールドの同じ行の直前にラベルがない場合、または右側のラベルが左側のラベルよりテキスト フィールドに近い場合、テキスト フィールドの右側にあるラベルが使用されます。
StartRow	この属性は記録されていませんが、手動でロケータに追加することができます。 StartRow を使用して、この行で始まるテキスト入力フィールド、 RumbaTextField を識別します。
StartColumn	この属性は記録されていませんが、手動でロケータに追加することができます。 StartColumn を使用して、この列で始まるテキスト入力フィールド、 RumbaTextField を識別します。
すべての動的ロケータ属性。	動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

Rumba での画面検証の使用

Rumba に対する画面検証を自動的に挿入するには、**オプション** ダイアログ ボックスで **記録 > 全般 > 画面検証を記録する** をオンにします。

画面検証を手動で挿入するには、以下を実行します。


1. テストで、**検証タイプのロジックの作成** ボタンをクリックし、**テスト ロジック デザイナ - 検証** を開きます。
2. **次へ** をクリックします。
3. **画面のコンテンツ** を選択します。
ツール > オプション > 記録 > Rumba > 除外オブジェクト で特定されるすべての除外オブジェクトが使用されます。この手順を完了後に、テストの **プロパティ** ウィンドウでこれらをさらにカスタマイズできます。
4. **次へ** をクリックします。
5. **識別** ボタンをクリックします。
6. 識別する Rumba 画面でコントロールを選択します。画面全体がキャプチャされます。
7. **次へ** をクリックします。
8. **完了** をクリックします。

Unix ディスプレイのテスト

Rumba の Unix ディスプレイは完全にテキストベースで、メインの **RUMBA 画面** コントロール以外に UI コントロールはありません。Unix ディスプレイ上のテストを再生するには、SendKeys メソッドを使用して、Unix ディスプレイにキーを送信します。Silk Test Workbench は、Unix ディスプレイ上での記録をサポートしません。

SAP のサポート

Silk Test Workbench は、Windows ベースの GUI モジュールを基にした SAP クライアント/サーバー アプリケーションのテストを組み込みでサポートしています。

 **注:** Silk Test Workbench のプレミアム ライセンスを所有している場合にのみ、Silk Test Workbench で SAP アプリケーションをテストできます。ライセンス モードについての詳細は、「ライセンス情報」を参照してください。

 **注:** Internet Explorer や Firefox 上から SAP NetWeaver を使用する場合、Silk Test Workbench は、xBrowser テクノロジ ドメインを使用してアプリケーションをテストします。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。


サポートするコントロール

SAP のテストで利用可能な記録および再生コントロールの完全な一覧については、「[SAP クラス リファレンス](#)」を参照してください。


サポートされている属性の一覧については、「[SAP アプリケーションの属性](#)」を参照してください。

SAP クラス リファレンス

SAP アプリケーションを設定すると、Silk Test Workbench は標準の SAP コントロールのテストのサポートを組み込みで提供します。

 **注:** Silk Test Workbench が SAP コントロールを解決できるようにするには、サーバー上で SAP GUI Scripting を有効にします。SAP GUI Scripting の有効化についての情報は、『[Introduction to SAP Gui Scripting](#)』を参照してください。SAP GUI Scripting が有効になっていないと、Silk Test Workbench は、すべてのコントロールを Win32 コントロール、およびカスタムコントロールとして解決します。

SAP クラス リファレンスに含まれるクラス (インクルードしたプロパティとメソッドを含む) は、Silk Test Workbench から直接アクセス可能な SAP オートメーション モジュールの一部です。


 **注:** インターフェイスに備わるアルゴリズムや動作を含み、インターフェイスは Silk Test Workbench のコントロール下にはありません。単純なデータ型が含まれているかぎり、ビジュアルテストはリストした SAP クラスのプロパティとメソッドを利用します。たとえば、SAPTable クラスの GetCell メソッドをビジュアルテストから呼び出すことはできませんが、VB .NET スクリプトからそのメソッドを呼び出すことができます。

SAP アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジ ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。


SAP がサポートする属性は次のとおりです。

- automationId
- caption

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ローケータ属性は、ワイルドカード ? および * をサポートしています。

SAP メソッドの動的な呼び出し

動的呼び出しを使用すると、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、メソッドの呼び出し、プロパティの取得、またはプロパティの設定を直接実行できます。また、このコントロールの Silk Test Workbench API で使用できないメソッドおよびプロパティも呼び出すことができます。動的呼び出しは、作業しているカスタム コントロールを操作するために必要な機能が、Silk Test Workbench API を通して公開されていない場合に特に便利です。

 **注:** 動的呼び出しは、スクリプトで使用できます。ビジュアル テストでは使用できません。


オブジェクトの動的メソッドは Invoke メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。


オブジェクトの複数の動的メソッドは InvokeMethods メソッドを使用して呼び出します。コントロールでサポートされている動的メソッドのリストを取得するには、GetDynamicMethodList メソッドを使用します。

動的プロパティの取得には GetProperty メソッドを、動的プロパティの設定には SetProperty メソッドを使用します。コントロールでサポートされている動的プロパティのリストを取得するには、GetPropertyList メソッドを使用します。

たとえば、テスト対象アプリケーション内のコントロールの実際のインスタンスに関して、タイトルを String 型の入力パラメータとして設定する必要がある setTitle というメソッドを呼び出すには、次のように入力します :

```
control.Invoke("SetTitle", "my new title")
```

 **注:** 通常、ほとんどのプロパティは読み取り専用で、設定できません。

 **注:** ほとんどのテクノロジー ドメインでは、メソッドを呼び出してプロパティを取得する場合、Reflection を使用します。

サポートされているメソッドおよびプロパティ

次のメソッドとプロパティを呼び出すことができます。

- Silk Test Workbench がサポートするコントロールのメソッドとプロパティ。
- SAP オートメーション インターフェイスによって定義されているすべての public メソッド
- コントロールが標準コントロールから派生したカスタム コントロールの場合、標準コントロールが呼び出すことのできるすべてのメソッドとプロパティ。

サポートされているパラメータ型

次のパラメータ型がサポートされます。

- すべての組み込み Silk Test Workbench 型

Silk Test Workbench 型には、プリミティブ型 (boolean、int、string など)、リスト、およびその他の型 (Point や Rect など) が含まれます。

- UI コントロール

UI コントロールは、TestObject として渡したり、返したりできます。

戻り値

プロパティや戻り値を持つメソッドの場合は、次の値が返されます。

- すべての組み込み Silk Test Workbench 型の場合は正しい値。これらの型は、「サポートされているパラメータ型」のセクションに記載されています。
- 戻り値を持たないすべてのメソッドの場合、C# では null が、VB では Nothing が返されます。

例

以下の VB .NET スクリプトの例では、SAP メソッドを動的に呼び出す方法を示します。

```
Dim _desktop As Desktop = Agent.Desktop
Dim wnd As SapWindow = _desktop.SapWindow("wnd 0")
Dim result As Object

result = wnd.Invoke("IsVKeyAllowed", 8) ' boolean return value
Console.WriteLine(
    "invoke result=" & result & " type=" & result.GetType().ToString())
wnd.Invoke("ShowMessageBox", "A Title", "Some text...", 1, 1)
```

SAP コントロールの動的呼び出し

Silk Test Workbench で SAP コントロールに対する操作を記録できない場合、SAP で利用できるレコーダーで操作を記録してから、記録されたメソッドを Silk Test Workbench スクリプトで動的に呼び出すことができます。これによって、記録できない SAP コントロールに対する操作を再生できます。

1. コントロールに対して実行する操作を記録するには、SAP で利用できる **SAP GUI スクリプト作成** ツールを使用できます。
SAP GUI スクリプト作成 ツールの詳細については、SAP のドキュメントを参照してください。
2. 記録された操作を **SAP GUI スクリプト作成** ツールによって保存された場所から開き、記録されたメソッドを確認します。
3. Silk Test Workbench で、記録されたメソッドをスクリプトから動的に呼び出します。

使用例

たとえば、SAP UI で *Test* というラベルが付いた、ボタンとリスト ボックスの組み合わせである特別なコントロールを押し、コントロールのサブメニュー *subsub2* を選択する操作を再生したい場合、SAP で利用できるレコーダーでこの操作を記録できます。結果のコードは、以下のようになります。

```
session.findById("wnd[0]/usr/ctrlCONTAINER/shellcont/shell").pressContextButton "TEST"
session.findById("wnd[0]/usr/ctrlCONTAINER/shellcont/shell").selectContextMenuItem "subsub2"
```

ここで、メソッド `pressContextButton` と `selectContextMenuItem` を Silk Test Workbench のスクリプトから動的に呼び出すには、次のコードを使用できます。

```
.SapToolBarControl("shell ToolbarControl").Invoke("pressContextButton", "TEST")
.SapToolBarControl("shell ToolbarControl").Invoke("selectContextMenuItem", "subsub2")
```

このコードを再生すると、SAP UI のコントロールが押下され、サブメニューが選択されます。

SAP の自動セキュリティ設定の構成

SAP アプリケーションを起動する前に、セキュリティ警告設定を構成する必要があります。このようにしないと、テストで SAP アプリケーションが再生されるたびにセキュリティ警告「スクリプトから GUI に接続しようとしています」が表示されます。

1. Windows の **コントロール パネル** で **SAP システム設定** を選択します。 **SAP システム設定** ダイアログ ボックスが開きます。
2. **デザイン選択** タブで、**スクリプトが実行中 SAP GUI に追加されるとき通知** をオフにします。

SAP eCATT の操作

Silk Test Workbench は、SAP の eCATT (*extended Computer Aided Test Tool*) と統合することにより、HTML テクノロジに基づく製品について完全なテスト サポートを SAP ユーザーに提供します。これにより、eCATT テスト スクリプト開発者は、HTML 製品のあらゆるテストを一元管理できます。Silk Test Workbench では、ANSI または Unicode データベースのデータの操作と保存ができます。

Silk Test Workbench を eCATT のユーザー インターフェイス内で使用すると、Silk Test Workbench の資産は eCATT テストと共に取得され、SAP/R3 データベースに格納されます。テスト担当者は、自分の結果の概要を一元的に把握できます。eCATT と Silk Test Workbench テストの結果は結合され、eCATT で同じ画面に表示できます。

開発者は、eCATT と Silk Test Workbench との間でパラメータを受け渡すことによって、真の意味で統合されたテスト シナリオを作成できます。また、テスト担当者は、SAP テストデータ コンテナのデータを Silk Test Workbench で使用できるようになります。最も重要な利点は、同じ SAP システムにアクセスできるテスト担当者がそれぞれのテストを共有できるということです。

Silk Test Workbench では、スクリプト引数を使用して、eCATT と Silk Test Workbench との間でパラメータを渡します。

Silk Test Workbench と eCATT 間のインターフェイスには、COM と RFC という 2 種類の通信が使用されます。eCATT における Silk Test Workbench の GUI インターフェイスは COM 経由で行われます。RFC 通信は、Silk Test Workbench が SAP/R3 データベースとの間でデータの保存と読み込みを行うために使用されます。データには、1 つまたは複数の Silk Test Workbench プロジェクトと、SAP/R3 データベースにラージ オブジェクトとして格納される、関連する Silk Test Workbench 資産のすべてが含まれます。これらのラージ オブジェクトのことを *BLOB* と呼びます。

Silk Test Workbench と SAP eCATT の統合

Silk Test Workbench と SAP eCATT の統合によって、テストプロセスを最初から最後まで管理できるようになります。一度 Silk Test Workbench と SAP eCATT を統合すれば、Silk Test Workbench と eCATT を次の 2 つの方法で使用することができます。


- eCATT 内から Silk Test Workbench を起動して、テスト スクリプトの記録、再生、デバッグ、および編集を行うことができます。また、資産を BLOB に保存したり、SAP/R3 データベースから BLOB を読み込んだり、テスト スクリプト引数を管理したりできます。こちらの方法を推奨します。
- Silk Test Workbench のスタンドアロン版を eCATT から別に開始します。Silk Test Workbench と eCATT 間で、データを BLOB で転送できます。

eCATT との統合のセットアップ

Silk Test Workbench と eCATT 間の統合を設定するには、以下の手順を実行します。

1. 永続的な変更を行うために、SAP サーバーに管理者としてログインします。
2. アプリケーション サーバーでスクリプトを有効化します。追加の情報については、SAP ヘルプのトピック「アプリケーションサーバでのスクリプト有効化」を参照してください。

- SAP サーバー上で、eCATT の外部ツールとして Silk Test Workbench を登録します。追加の情報については、SAP ヘルプのトピック「eCATT での外部ツール使用準備」を参照してください。
- 各クライアントに対して、クライアントのフロントエンドでスクリプトを有効化します。

 **注:** Silk Test Workbench が SAP アプリケーションに対して記録しようとするたびに確認ダイアログ ボックスが表示されることを避けるために、**実行中の GUI にスクリプトが接続する時に通知する** チェック ボックスをオフにします。

追加の情報については、SAP ヘルプのトピック「フロントエンドでのスクリプト有効化」を参照してください。

- SAP サーバー上で、新しいユーザー アカウントを作成します。
- eCATT と組み合わせて Silk Test Workbench を使用しているクライアント マシンに、Silk Test Workbench と SAPGUI クライアントの両方をインストールします。

eCATT への Silk Test Workbench の登録

Silk Test Workbench と eCATT を統合するには、*ECCUST_ET* SAP テーブルに Silk Test Workbench を登録する必要があります。SET_EXTERNAL_TOOL メソッドは、*ECCUST_ET* テーブルに必要な項目を作成します。SET_EXTERNAL_TOOL 汎用モジュールは SE37 トランザクションを使用して呼び出すことができます。

- 最初の画面で、汎用モジュール名 SET_EXTERNAL_TOOL. を入力します。
- 汎用モジュール** メニューから、**テスト > 単一テスト** を選択します。
- 次のウィンドウで、次のパラメータと値を入力します。

パラメータ	値
TOOL_NAME	Silk Test Workbench
PROG_ID	SilkTest.STWEcattControl
TOOL_DESC	Silk Test Workbench for eCATT
TOOL_DATABASE	STW_EDIT
TOOL_RUN_DB	STW_EXEC
TOOL_NO_PWD	<blank>
TOOL_NO_DB	<blank>

- 実行** ボタン (F8) を押します。

特定のユーザー アカウントの作成

外部ツールを使用して eCATT 統合を利用するために、システム管理者はシステムに標準ユーザーを生成する必要があります。これは、SE38 の ECATT_GENERATE_ET_USER プログラムを各システムに対して一度だけ実行することによって行われます。

レポートを実行した後に、新しく作成したユーザー ロールをアクティブにするには、次の手順を実行します。

- PFCG トランザクションでロール SAP_ECET を入力し、**Change** を選択します。表示される警告は無視してかまいません。
- Authorizations** タブに切り替えて、**Change Authorization Data** を選択します。
- SAP_ECET** ツリーのトップ ノード上にカーソルを移動し、**Authorizations > Generate** を選択します。
- Back** をクリックして、ロールのメンテナンス画面に戻ります。
- Save** をクリックします。

Silk Test Workbench のスタンドアロン版の構成

このセクションのトピックでは、Silk Test Workbench のスタンドアロン版を eCATT からとは別に開始した場合に、Silk Test Workbench と eCATT 間で BLOB でデータを転送したい場合に、行わなければならない追加の構成について説明します。Silk Test Workbench を eCATT 内から開始した場合は、この設定を行う必要はありません。

SAP/R3 ログイン パラメータの設定

SAP/R3 ログイン パラメータは、SAP/R3 データベースにログオンするために使用されます。Silk Test Workbench を使用して SAP/R3 データベースにログオンし、Silk Test Workbench と eCATT の間で BLOB の取得や転送ができます。

1. 以下のいずれかを実行します。

- a) **BLOB の保存** ダイアログ ボックスから **ログイン パラメータの変更** ボタンをクリックします。
- b) Silk Test Workbench のメニューから **ツール > SAP eCATT > SAP/R3 ログイン パラメータ** をクリックします。

SAP/R3 ログイン ダイアログ ボックスが表示されます。eCATT に接続している場合は、設定されたログオン パラメータが表示されます。この場合、パスワードを再入力するだけでログオンできます。すでに接続している場合は、ステップ 3 に進んでください。

現在 eCATT に接続していない場合、SAP eCATT データベースにワンタイム ログオンを実行する必要があります。SAP ではログオン情報が維持されるため、一度ログインしたあとは、パスワードを入力するだけで再接続してデータを転送できます。

2. 3 桁の SAP 固有のクライアント セッション番号を **クライアント フィールド**に入力します。クライアント番号の詳細については、SAP SDK のマニュアルを参照してください。
3. SAP/R3 データベースのログオン ユーザー ID を **ユーザー名 フィールド**に入力します。
4. SAP/R3 のパスワードを **パスワード フィールド**に入力します。
5. 有効な SAP 言語コードを **言語 フィールド**に入力します。
6. **OK** をクリックして接続するか、**キャンセル** をクリックして接続せずに終了します。

RFC 接続パラメータの設定または変更

RFC インターフェイスは、Silk Test Workbench と SAP/R3 データベース間で BLOB、引数コンテナ、および実行情報を転送するのに使用されます。Silk Test Workbench と SAP eCATT の間で BLOB の保存と読み込みを行うには、接続情報が必要となります。以下の手順に従って、RFC インターフェイスの接続パラメータを変更してください。

1. eCATT インターフェイスを使用している場合は、Silk Test Workbench のアクティブ セッションに切り替えます。
2. 以下のいずれかを実行します。
 - a) **BLOB の保存** ダイアログ ボックスから **RFC パラメータの変更** ボタンをクリックします。
 - b) Silk Test Workbench のメニューから **ツール > SAP eCATT > RFC 接続パラメータ** をクリックします。

RFC 接続パラメータ ダイアログ ボックスが表示されます。

3. 以下のいずれかを実行します。

現在のセッションに対して RFC 接続をすでに確立してある場合は、**RFC 接続パラメータ** ダイアログ ボックスにパラメータが表示されます。これらのパラメータを変更する必要がない場合は、ステップ 4 へ進んでください。

現在のセッションに対して RFC 接続をまだ確立していない場合や、RFC 接続パラメータを変更する必要がある場合は、必要な接続情報を入力します。接続情報は以下のとおりです。

SAP ログオン ID を使用する	すでに SAP GUI を使用して SAP ログオン ID を作成してある場合は、このボックスをオンにしてログオンを選択します。SAP GUI に項目がない場合は、このチェックボックスと SAP ログオン ID は無効になっています。
SAP ログオン ID	リストからログオンを選択します。
AS ホスト	RFC 接続に使用される SAP アプリケーション サーバーのホスト マシンの名前。これには、ホスト名の前に付く SAP ルーティング文字列を含めることができます。
RFC の種類	Silk Test Workbench の SAP eCATT への RFC 接続に使用されるサーバーの種類。R3 システムの場合、RFC の種類は 3 です。
システム番号	RFC 接続の SAP システム番号。

- SAP サーバーが SAP GUI クライアントおよび Silk Test Workbench 以外の別のネットワーク上にある場合は、SAP ルーター文字列を eCATT ABAB コードに手動で入力する必要があります。以下のステップを実行します。
 - トランザクション se80 を開始します。
 - Class/Interface cl_apl_ecatt_extprog を編集します。
 - メソッド Launch に移動して次のコードを追加します： `ashost = '<ROUTER-STRING+IP>'`。
ここで、<ROUTER-STRING+IP> は SAP ルーター文字列、IP は SAP サーバーのアドレスです。
- OK** をクリックして変更を適用し、ダイアログ ボックスを閉じます。BLOB の保存と SAP/R3 のログインが Silk Test Workbench と eCATT の間で行われると、BLOB が SAP/R3 データベースに転送されます。BLOB が転送されたら、そのテスト スクリプトを eCATT で使用できるようになります。

Silk Test Workbench を eCATT から開始した場合の eCATT スクリプトの記録とテスト

Silk Test Workbench を eCATT から開始した場合、eCATT を使用してテストを定義し、Silk Test Workbench を eCATT の外部テスト ツールとして使用してスクリプトを記録することができます。これにより、テストを一元管理でき、Silk Test Workbench 内のスクリプト作成機能を使用してターゲット アプリケーションの記録とテストを実行できるようになります。Silk Test Workbench で記録されるテストは、.NET スクリプトかビジュアル テストのいずれかになります。

eCATT でのテスト スクリプトの定義

- テスト スクリプトを eCATT で起動します。それには、eCATT の最初の画面で **テスト スクリプト** を選択し、名前として S_ECATT を入力し、バージョン番号を入力します。
- eCATT で、Silk Test Workbench を外部ツールとして指定します。
- ツールバーの **オブジェクトの適用** ボタンをクリックします。SAP eCATT でスクリプト エディタが表示されます。

eCATT でのテスト スクリプト属性の指定

- eCATT を使用してテスト スクリプト属性を指定します。詳細な手順については、<http://help.sap.com> にある eCATT ヘルプを参照してください。
- アプリケーション ツールバーの **スクリプト** ボタンをクリックします。**外部ツールのユーザエントリ** ダイアログ ボックスが表示されます。
- ST_EDIT データベースのユーザー名とパスワードを入力します。
これは、Silk Test Workbench を外部ツールとして使用するために必要です。
- テスト スクリプトの作成場所とする外部ツール プロジェクトの名前を入力します。この名前は、Silk Test Workbench 内のプロジェクトの名前です。
- ST_EDIT データベースをクリアしたい場合は、チェック ボックスをオンにします

eCATT から Silk Test Workbench が起動され、eCATT で作成された新しいテスト スクリプトが **コード** ウィンドウに表示されます。これで、テスト スクリプトを記録してテストできます。または、**コード** ウィ



ンドウを閉じて、ビジュアルテストを作成および記録して、ターゲットアプリケーションをテストすることができます。

Silk Test Workbench を使用したスクリプトの記録とテスト

1. ターゲットアプリケーションが実行されており、使用可能な状態であることを確認します。
2. Silk Test Workbench で、.NET スクリプトを記録するか、ビジュアルテストを作成して記録します。
3. 省略可能：任意の引数を作成します。
詳細については、*Silk Test Workbench* における eCATT スクリプト引数のサポートを参照してください。
4. Silk Test Workbench を使用して .NET スクリプトまたはビジュアルテストを再生します。再生用に作成された結果を表示することもできます。
5. テストおよび関係するすべての Silk Test Workbench 資産を BLOB に保存します。保存された BLOB データは SAP/R3 データベースに転送され、eCATT で使用できます。

スタンドアロン Silk Test Workbench を使用した eCATT スクリプトの作成

Silk Test Workbench で eCATT スクリプトを作成し、追加の VB.NET コードを記録または記述してテストしたあと、そのテストスクリプトを eCATT で使用できるように SAP/R3 データベースに転送できます。この場合は、BLOB でテストスクリプト、プロジェクト、および関連資産が eCATT に渡されるまで、Silk Test Workbench はスタンドアロン環境で実行されます (eCATT から起動しません)。

1. Silk Test Workbench を起動します。
2. テストスクリプトを作成して記録します。
 **注:** eCATT で作成したテストスクリプトを BLOB で Silk Test Workbench データベースの一時記憶域に読み込み、使用することもできます。
3. テストスクリプトを再生します。
4. テストスクリプトの再生用に作成された結果をすべて表示します。
5. eCATT でのテストスクリプトに必要な引数をすべて作成します。
6. テストスクリプトとそのプロジェクト、および関連資産を eCATT に転送するには、テストスクリプトを BLOB に保存します。Silk Test Workbench はスタンドアロンモードで実行されているので、[SAP/R3 ログイン] ダイアログボックスが表示されます。
7. SAP/R3 データベース ログイン パラメータを設定し、BLOB 転送を行うための SAP/R3 データベースに接続します。
 **注:** Silk Test Workbench では SAP/R3 データベースに接続するのに有効な SAP/R3 ログイン ユーザー名とパスワードが必要です。具体的な手順については「SAP/R3 ログインパラメータの設定」を参照してください。
8. 現在のセッションに対して RFC 接続を確立する必要があります。BLOB データを eCATT に転送するために RFC 接続パラメータを設定します。接続が確立されると、eCATT でテストスクリプトを使用できるようになります。

BLOB の操作

バイナリ ラージ オブジェクト (BLOB) は、SAP eCATT と Silk Test Workbench との間でテストスクリプトまたはビジュアルテストを転送するのに使用されるコンテナです。BLOB 内の情報には、.NET スクリプトまたはビジュアルテストに属するすべての関連テスト資産 (結果を除く) が含まれ、他の .NET スクリプトまたはビジュアルテストも含まれます。

BLOB には、関連テスト資産以外に、テストスクリプトまたはビジュアルテストに属するすべてのオプション設定も含まれます。

BLOB は XML ドキュメントです。BLOB の形式は、Silk Test Workbench の内部でのみ使用されます。BLOB のデータ自体は、外部のシステムによって解釈することはできません。

最後に保存したバージョンの BLOB だけが SAP eCATT データベースに保存されます。以前のバージョンは保持されません。BLOB がユーザーによる編集/デバッグのために開かれると、この BLOB はデータベースに再送信されるまで別のユーザーによってアクセスすることはできません。BLOB を再送信すると、以前のバージョンは上書きされます。


BLOB は、Silk Test Workbench で作成 (保存) されます。BLOB を保存すると、その BLOB は、Silk Test Workbench と eCATT 間の RFC インターフェイスを介して、Silk Test Workbench から SAP/R3 データベースに転送されます。

BLOB 追跡情報の入力

1. 開いているテスト スクリプトのオプションがすべて適切に設定されていることを確認します。特に、Silk Test Workbench の再生オプションは eCATT での再生に影響を与えます。
2. Silk Test Workbench のメニューから以下のいずれかをクリックします。
 - a) BLOB を保存して Silk Test Workbench での作業を続けるには、**ツール > SAP eCATT > BLOB の保存 > 保存して続行** をクリックします。
 - b) BLOB を保存して BLOB が SAP/R3 データベースに転送されたときに eCATT に移動するには、**ツール > SAP eCATT > BLOB の保存 > 保存して eCaTT へ移動** をクリックします。保存が完了すると、Silk Test Workbench が閉じ、eCATT にフォーカスが設定されます。

BLOB の保存 ダイアログ ボックスが表示されます。このダイアログ ボックスには、最後に読み込まれた BLOB に関する情報が表示されます。これらの情報を入力し直して、新しい BLOB を保存できます。


3. BLOB の一意の名前を [BLOB ID] フィールドに入力します。
4. BLOB のバージョン番号を [バージョン] フィールドに入力することもできます。バージョンを指定しない場合は、「バージョン 1」が使用されます。

 **注:** バージョン番号は検証も増分もされません。現在のバージョンの BLOB だけが SAP/R3 データベースに保存されます。

5. 診断データをローカルの XML ファイルに保存する場合は、[ファイルにダンプ] チェック ボックスをオンにします。
6. BLOB の内容を入力して BLOB の情報の入力を続けます。

BLOB の内容の入力または変更

エクスポートするプロジェクト リストには、Silk Test Workbench データベースにあるプロジェクトのうち自分がアクセス権を持つものがすべて表示されます。

 **注:** **接続情報** セクションは、スタンドアロン版の Silk Test Workbench を使用しているときのみ表示されます。eCATT から Silk Test Workbench を開始した場合には表示されません。

1. 該当するチェック ボックスをオンにして、BLOB に挿入するプライマリ テスト スクリプトを含むプロジェクトを選択します。他のテスト スクリプトに対する呼び出しがプライマリ スクリプトに含まれている場合は、それらのプロジェクトも選択します。プロジェクトを選択すると、定義済みの記録および再生設定グループのすべてが **[記録設定]** および **[再生設定]** リストに表示されます。これらのグループは、Silk Test Workbench の記録および再生のオプションで設定されます。
2. 定義済みの記録オプションのグループを BLOB に含めるには、**記録設定** リスト ボックスからその記録設定グループを選択します。
3. 定義済みの再生オプションのグループを BLOB に含めるには、**再生設定** リスト ボックスからその再生設定グループを選択します。
4. Silk Test Workbench のプライマリ プロジェクトを **プライマリプロジェクト** ボックスから選択します。プライマリ プロジェクトとは、BLOB のプライマリ スクリプトが属する Silk Test Workbench プロジェクトのことです。
5. **資産の種類** ボックスで、BLOB に保存されるテストの種類を決めます。.NET スクリプトまたはビジュアル テストをリストから選択します。
6. **プライマリ テスト スクリプト/ビジュアル テスト** ボックスには、選択したプロジェクトにある使用可能なテスト スクリプトまたはビジュアル テストのすべてが表示され、現在読み込まれている BLOB が

らの .NET スクリプトまたはビジュアル テスト (これらは一時的に Silk Test Workbench データベースに保存されます) も表示されます。[資産の種類] での選択に応じて、.NET スクリプトまたはビジュアル テストがリストに表示されます。この BLOB のプライマリ .NET スクリプトまたはビジュアル テストをリストから選択します。

7. **保存** をクリックして BLOB を SAP/R3 データベースに保存します。



注:

デフォルトでは、STW_EDIT および STW_EXEC の Silk Test Workbench の DSN は大文字ですが、SAP サーバーにインストールされている DSN の大文字/小文字と一致させる必要があります。たとえば、BLOB の保存時に「STW_Edit が使用できません」というエラーが表示された場合は、DSN 名の大文字/小文字を確認してください。場合によっては、Silk Test Workbench の DSN を「STW_EDIT」から「STW_Edit」に変更する必要があります。

Silk Test Workbench のデフォルトの再生または記録オプションを変更する場合、設定の名前がなくなります。BLOB を保存するには、このような名前のない設定に対して名前を付けることが強制されます。再生と記録の設定は「プロファイル」に保存できます。詳細については、「再生オプションの保存」および「記録オプションの保存と再利用」を参照してください。

Silk Test Workbench で [SAP/R3 ログイン] ダイアログ ボックスが表示され、データベース ログインパラメータを設定でき、BLOB 転送を行うための SAP/R3 データベースに接続できます。

SAP/R3 データベースにすでに接続している場合は、Silk Test Workbench でデータベースのパスワードを入力するだけで済みます。まだ SAP/R3 データベースに接続していない場合は、Silk Test Workbench で初回接続時に SAP/R3 のログイン ユーザー名とパスワードを入力する必要があります。具体的な手順については「SAP/R3 ログインパラメータの設定」を参照してください。

RFC 接続がすでに確立されている場合は、**AS ホストとクライアントの情報** が **BLOB の保存** ダイアログボックスに表示されます。これらの情報が表示されない場合は、現在のセッションで RFC 接続は確立されていません。その場合、BLOB データを eCATT に転送するには、RFC 接続パラメータを設定する必要があります。「RFC 接続パラメータの設定または変更」を参照してください。

Silk Test Workbench における eCATT スクリプト引数のサポート

eCATT スクリプト引数は、eCATT 環境における重要な概念の 1 つです。これらの引数は、eCATT と Silk Test Workbench テストとの間の「インターフェイス」となります。引数には、eCATT で BLOB テスト スクリプトまたはビジュアル テストを実行するために現在の BLOB で必要な追加のデータが含まれています。これらの引数によって、テスト スクリプトやビジュアル テストを eCATT 環境で使用できるようになります。

引数は、Silk Test Workbench または eCATT で作成することができ、両方の環境の間でインポートしたりエクスポートしたりできます。

Silk Test Workbench における引数のサポートは、以下の要素で構成されます。

Silk Test Workbench の引数メンテナンス機能

Silk Test Workbench の引数メンテナンス機能では、**eCATT 引数コンテナ** ダイアログボックスを使用して、新しい引数を作成したり、既存の引数を編集または削除したりできます。**eCATT 引数コンテナ** ダイアログボックスを使用して、BLOB を介して eCATT スクリプトとの間でインポートまたはエクスポートできる引数を定義します。定義した引数は、Silk Test Workbench/CATT COM インターフェイスを使用して、eCATT スクリプトと Silk Test Workbench .NET スクリプトまたはビジュアル テストとの間で受け渡すことができます。

Silk Test Workbench は、一度に 1 つの BLOB (引数と関連資産を持つメイン スクリプト) しか処理できないので、**eCATT 引数コンテナ** ダイアログボックスには常に現在の BLOB のスクリプト引数が表示されます。

Silk Test Workbench では、BLOB を SAP データベースに格納するとき、BLOB の引数に指定された値を読み取り、引数コンテナで転送して、BLOB と共に格納しま

す。Silk Test Workbench の引数メンテナンス機能は文字列データ型スクリプト引数でのみ機能します。

eCATT スクリプト引数を使用する

Silk Test Workbench .NET スクリプトの場合は、Silk Test Workbench のスクリプト言語にある 2 つのメソッドを使用して、スクリプトとの間で引数値が受け渡されます。Workbench.SetECATTArg() メソッドは、SAP eCATT スクリプトに渡す引数値を作成します。Workbench.GetECATTArg() メソッドは、SAP eCATT 引数コンテナから渡された引数値を取得します。

ビジュアル テストの場合は、eCATT 引数の設定 変数ステップを作成して、定義された引数値を eCATT スクリプトにエクスポートします。eCATT 引数の内容の取得 変数ステップを作成して、定義された引数値を eCATT スクリプトからインポートしてローカル変数に保存し、その値をビジュアル テストで使用できます。

eCATT スクリプト引数の作成

1. BLOB が読み込まれている間、**ツール > SAP eCATT > 引数コンテナ** をクリックします。**eCATT 引数コンテナ** ダイアログ ボックスが表示されます。現在読み込まれている BLOB に対して定義された引数がデータ テーブルに表示されます。
2. **追加** をクリックします。**引数の追加** ダイアログ ボックスが表示されます。
3. **名前** フィールドに、引数に付ける名前を入力します。
4. 引数の転送の方向を選択します (**方向** フィールドのリストから **EXPORT** または **IMPORT** を選択します)。
5. **デフォルト値** フィールドに、引数のデフォルト値を入力します。デフォルト値は、eCATT でテスト スクリプトによって使用されるデータです。デフォルト値の詳細については、eCATT のマニュアルを参照してください。
6. **値** フィールドに、引数の実際の値を入力します。この値はテストスクリプトから取得でき、eCATT で表示可能です。
7. 引数の説明を **説明** フィールドに入力することもできます。説明は、引数の名前がわかりにくい場合にそれを識別しやすくするためのものです。
8. **追加** をクリックして、現在の BLOB の引数コンテナに引数を追加します。**eCATT 引数コンテナ** ダイアログ ボックスで、引数がコンテナ内の項目として表示されます。
9. 他の引数を追加するには、ステップ 1 から 9 を繰り返します。
- 10 現在読み込まれている BLOB に対して必要な引数の作成をすべて終わったら、**閉じる** をクリックして **eCATT 引数コンテナ** ダイアログ ボックスに戻ります。
- 11 **OK** をクリックします。

eCATT スクリプト引数の編集

最近読み込まれた BLOB のプライマリ .NET スクリプトまたはビジュアル テストの引数は、まだ読み込まれている間に変更することができます。現在読み込まれている BLOB の eCATT スクリプト引数を編集するには、Silk Test Workbench の **eCATT 引数コンテナ** ダイアログ ボックスを使用します。

1. 引数を変更する BLOB が読み込まれている間、**ツール > SAP eCATT > 引数コンテナ** をクリックします。**eCATT 引数コンテナ** ダイアログ ボックスが表示されます。このダイアログ ボックスのデータ テーブルには、現在読み込まれている BLOB に対して定義された引数がすべて表示されます。
2. パラメータを編集する引数を選択します。
3. **編集** をクリックします。**引数の編集** ダイアログ ボックスが表示されます。
4. 必要に応じて引数の値を変更します。

名前 引数の名前。

- 方向** 引数が「IMPORT」（eCATT から Silk Test Workbench へ）なのか「EXPORT」（Silk Test Workbench から eCATT へ）なのかを決めます。
- デフォルト値** 引数のデータのデフォルト値です。デフォルト値は、eCATT スクリプト引数に値が指定されていない場合に、eCATT でテスト スクリプトによって使用されるデータです。デフォルト値の詳細については、eCATT のマニュアルを参照してください。
- 値** 引数の実際の値を入力します。この値はスクリプトから取得でき、eCATT で表示可能です。
- 説明** 説明は、引数の名前がわかりにくい場合にそれを識別しやすくするためのものです。

5. **編集** をクリックします。引数のパラメータが **eCATT 引数コンテナ** ダイアログ ボックスに表示されます。
6. 引数コンテナの他の引数を編集する場合は、ステップ 1 と 2 を繰り返します。
7. 現在読み込まれている BLOB に対して必要な引数の編集をすべて終わったら、**閉じる** をクリックして **eCATT 引数コンテナ** ダイアログ ボックスに戻ります。
8. **OK** をクリックします。

引数コンテナからの eCATT 引数の削除

最近読み込まれた BLOB のスクリプト引数を、いつでもその引数コンテナから削除することができます。ただし、Silk Test Workbench では、引数を削除する前に確認のメッセージが表示されないので注意してください。

1. BLOB が読み込まれている間、**ツール > SAP eCATT > 引数コンテナ** をクリックします。**eCATT 引数コンテナ** ダイアログ ボックスが表示されます。このダイアログ ボックスのデータ テーブルには、最後に読み込んだ BLOB に対して定義された引数がすべて表示されます。各引数が別々の行に表示されます。
2. 削除する引数がある行を選択して **削除** をクリックします。引数を含む行が引数コンテナから削除されます。
3. 削除する他の引数に対してステップ 2 を繰り返します。
4. **OK** をクリックします。

eCATT スクリプトからビジュアル テストへの引数のインポート

引数コンテナから eCATT スクリプト引数をインポートして、任意のビジュアル テストで引数の値を使用できます。渡された引数の内容は、ビジュアル テストのローカル変数に保存されます。

ビジュアル テストで eCATT スクリプトから引数をインポートするには、以下を実行します。

1. eCATT スクリプト引数をインポートするビジュアル テストを開きます。
2. どの引数を eCATT スクリプトからインポートするかを決定します。
3. Silk Test Workbench の **eCATT 引数コンテナ** ダイアログ ボックスで、ビジュアル テストで使用できる eCATT スクリプト インポート引数を作成します。
4. ビジュアル テストで、インポートする引数を保存するローカル変数を作成します。
定義したローカル変数はビジュアル テストの <<開始>> ステップに表示されます。ただし、変数は任意のテスト ステップに挿入できます。
5. ビジュアル テストでテスト ステップを選択し、**挿入 > SAP eCATT > 引数の取得** をクリックします。Silk Test Workbench によって、選択したステップの直後にステップが作成されます。この新しく作成されたステップは、eCATT 引数の値を取得し、この値を変数に格納するのに使用されます。これにより、この値をビジュアル テストで使用できるようになります。ステップのプロパティが **プロパティ** ペインに表示されます。



ヒント: テスト ステップの最適な構成を保つため、このテスト ステップをビジュアル テストの開始またはその近くに配置します。

6. テスト ステップの **プロパティ** ペインで、**並べ替えカテゴリ** アイコンをクリックして、プロパティをカテゴリごとにグループ化します。

7. 割り当てカテゴリ プロパティを更新します。

ステップの **割り当て** プロパティでは、インポートする eCATT 引数と、引数の値を格納するローカル変数を指定します。

- a) **eCATT 変数名** プロパティに、ビジュアルテストで使用する値を持つ eCATT スクリプト引数の名前を入力します。

指定した eCATT スクリプト引数は、ビジュアルテストの **eCATT 引数コンテナ** で定義した引数名の 1 つと一致しなければなりません。

- b) **Local variable name** プロパティには、インポートした引数を格納するために作成した変数を含む、ビジュアルテストに対して定義されているすべてのローカル変数がリストされます。指定した eCATT スクリプト引数の値を格納するローカル変数をリストから選択します。

eCATT 引数 '[eCATT スクリプト引数名]' の内容を取得して、ローカル変数 '[ローカル変数名]' に格納します

8. 割り当てカテゴリ プロパティを更新してビジュアルテストを保存すると、テストステップの記述は次のようになります。

```
Get contents of eCATT argument '[eCATT script argument name]' and put into local variable '[local variable name]'
```

9. 省略可能：複数の eCATT スクリプト引数をビジュアルテストにインポートする場合は、手順を繰り返してください。

eCATT スクリプト引数がビジュアルテストにインポートされ、ローカル変数に保存されると、これらのローカル変数は自動ステップの入力値として使用できるようになります。たとえば、ログイン ID に対する eCATT スクリプト引数の値を、ログイン ID フィールドに対して値を入力する自動ステップのテキスト入力として使用することができます。

ビジュアルテストから eCATT スクリプトに引数をエクスポートする

ビジュアルテストで使用されている引数を eCATT スクリプトに渡すことができます。eCATT スクリプト引数は BLOB に保存する必要があり、ビジュアルテストの引数コンテナ内でエクスポート引数として定義する必要があります。

eCATT スクリプト引数をビジュアルテストから eCATT スクリプトにエクスポートするには、以下を実行します。

1. 引数を eCATT スクリプトにエクスポートするビジュアルテストを開きます。
2. どの引数を eCATT スクリプトにエクスポートするかを決定します。
Silk Test Workbench の **eCATT 引数コンテナ** ダイアログボックスで、eCATT スクリプトエクスポート引数を作成します。
3. ビジュアルテストで **<<開始>>** ステップを選択し、**挿入 > SAP eCATT > 引数の設定** をクリックします。

Silk Test Workbench は、eCATT 引数のエクスポート値を設定するステップを作成します。これにより、この値を eCATT スクリプトで使用できるようになります。ステップのプロパティが **プロパティ** ペインに表示されます。



ヒント: テストステップの最適な構成を保つため、このテストステップをビジュアルテストの開始またはその近くに配置します。

4. ステップの **プロパティ** ペインで、**並べ替えカテゴリ** アイコンをクリックして、プロパティをカテゴリごとにグループ化します。
5. [割り当てカテゴリ] プロパティを更新して、eCATT 引数の名前とそれに割り当てる値を定義します。
 - a) **eCATT 変数名** プロパティで、値をエクスポートする eCATT スクリプト引数の名前を入力します。
指定する eCATT スクリプト引数は、ビジュアルテストの **eCATT 引数コンテナ** 内で定義されている引数名と一致している必要があります。
 - b) **値** プロパティで、引数の値を入力するか、値の領域を右クリックしてリテラル、変数、式の結果、または ActiveData 値を割り当てます。

[割り当てカテゴリ] プロパティを更新してビジュアル テストを保存すると、ステップは次のようになります。

```
Set eCATT argument '[eCATT script argument name]' to '[value]'
```

6. 省略可能：ビジュアル テストから eCATT スクリプトに複数の引数をエクスポートする場合は、この手順を繰り返してください。

引数がエクスポートされ、BLOB が保存されると、eCATT スクリプトで引数値を使用できます。

eCATT スクリプトから .NET スクリプトへの引数のインポート

引数コンテナから eCATT スクリプト引数を取得し、Silk Test Workbench の任意の VB .NET スクリプトで引数の値を使用します。

eCATT スクリプトから Silk Test Workbench の VB .NET スクリプトへ引数をインポートするには、以下を実行します。

1. eCATT スクリプト引数をインポートする VB .NET スクリプトを開きます。
2. どの引数を eCATT スクリプトから VB .NET スクリプトにインポートするかを決定します。
Silk Test Workbench の **eCATT 引数コンテナ** ダイアログ ボックスで、VB .NET スクリプトで使用できる eCATT スクリプト インポート引数を作成します。
3. .NET スクリプトで、インポートする引数を保存するローカル変数を作成します。
例：

```
Dim argName As String
```
4. Workbench クラスの GetECATTArg 関数を使用して、eCATT スクリプト引数を取得します。
例：

```
ArgName = Workbench.GetECATTArg("NAME")
```
5. 省略可能：GetECATTArg 関数は文字列を返します。.NET スクリプトで型キャストを使用して、日付など他の型に変更できます。

eCATT スクリプト引数が .NET スクリプトにインポートされると、引数は自動ステップの入力値として使用できるようになります。たとえば、ログイン ID に対する eCATT スクリプト引数の値を、ログイン ID フィールドに対して値を入力する自動ステップのテキスト入力として使用することができます。

.NET スクリプトから eCATT スクリプトに引数をエクスポートする

Silk Test Workbench の VB .NET スクリプトで使用した引数を eCATT スクリプトに渡して、スクリプトで引数を使用できます。

eCATT スクリプト引数を .NET スクリプトから eCATT スクリプトにエクスポートするには、以下を実行します。

1. 引数を eCATT スクリプトにエクスポートする .NET スクリプトを開きます。
2. どの引数を eCATT スクリプトにエクスポートするかを決定します。
3. Silk Test Workbench の **eCATT 引数コンテナ** ダイアログ ボックスで、.NET スクリプトで使用できる eCATT スクリプト エクスポート引数を作成します。
4. Workbench クラスの SetECATTArg 関数を使用して、eCATT スクリプト引数をエクスポートします。
例：

```
Workbench.SetECATTArg("OUTPUTNAME","TestName")
```

この例は、スクリプト引数 *TestName* を .NET スクリプトから eCATT のスクリプト引数 *OUTPUTNAME* にエクスポートしています。

 **注：** 指定した eCATT スクリプト引数は、**eCATT 引数コンテナ** で定義した引数名の 1 つと一致しなければなりません。

eCATT から Silk Test Workbench でテストを実行する


ビジュアルテストまたは .NET スクリプトを eCATT で実行するには、ビジュアルテストまたは .NET スクリプトを含む BLOB を作成し保存する必要があります。**BLOB の保存** ダイアログボックスの **BLOB コンテンツ** 領域にある **プライマリテストスクリプト/ビジュアルテスト** 設定によって、どのテストを実行するかを決定するために eCATT によって使用されます。

ビジュアルテストまたは .NET スクリプトを Silk Test Workbench で実行する eCATT でスクリプトを実行するには：

1. テストを含んだ BLOB が保存されていない場合、BLOB を保存します。Silk Test Workbench で、**ツール > SAP eCATT > BLOB の保存 > 保存して eCATT へ移動**。**BLOB の保存** ダイアログボックスの **BLOB コンテンツ** 領域にある **プライマリテストスクリプト/ビジュアルテスト** 設定によって、どのテストを実行するかを決定するために eCATT によって使用されます。
2. eCATT で **オブジェクトディレクトリエントリ登録** ダイアログ 僕巢が開きます。**パッケージ** の名前を入力し、**保存** をクリックします。以上でスクリプトを eCATT UI で実行する準備が整いました。
3. eCATT UI で **実行** をクリックします。**開始オプション** ダイアログボックスが開きます。
4. **外部ツール** 領域で、Silk Test Workbench データベースの **ユーザー名** と **パスワード** を入力します。
5. 省略可能：テストが実行されている間に Silk Test Workbench の UI で何が起きているのか確認したい場合は、**外部ツールの開始モード** リストボックスから**外部ツールのユーザインタフェースを使用して実行** を選択します。
リストボックスから **デバッグモード** を選択した場合、スクリプトは最初の行で停止します。
6. **実行** をクリックします。eCATT は Silk Test Workbench を開始し、情報を送信してスクリプトを実行します。

eCATT で Silk Test Workbench テスト結果を表示する

Silk Test Workbench でビジュアルテストまたは .NET スクリプトを実行して生成されたテスト結果を eCATT で表示することができます。eCATT が Silk Test Workbench でビジュアルテストまたは .NET スクリプトを実行すると、ログが eCATT に作成されます。このログには、Silk Test Workbench テストの結果が含まれています。

 **注：** Silk Test Workbench テスト結果に含まれるスクリーンショットは、ログには含まれません。


ユニバーサル Windows プラットフォームのサポート

ユニバーサル Windows プラットフォーム API を使った Microsoft Windows 10 アプリのテストのサポートについて説明します。

UWP は Microsoft によって作成された API で、Microsoft ストアから購入、ダウンロードして利用できるアプリを開発できます。

Silk Test Workbench では、次のオペレーティングシステムにおける UWP アプリのテストを組み込みでサポートします。

- Microsoft Windows 10
- Microsoft Windows Server 2019

 **注：** Silk Test は、Microsoft Windows 8 および Microsoft Windows 8.1 上でのユニバーサル Windows プラットフォーム (UWP) アプリ (Windows Store アプリ、または Metro スタイルアプリともいう) のテストをサポートしません。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

サポートするコントロール

Silk Test Workbench では、UIA コントロールをベースにしたコントロールに加え、以下のコントロールに対する記録と再生をサポートします。

- UWPCoboBoxItem
- UWPCoboBox

アプリケーション構成

UWP アプリのアプリケーション構成では、以下を設定します。

実行可能ファイル UWP アプリの実行可能ファイルは、アプリの名前、パブリッシャー ID、およびパッケージのコンテンツでアプリを一意に識別するアプリ識別子の一部を組み合わせた文字列です。たとえば、UWP アプリの名前が *Microsoft.WindowsCalculator* で、パブリッシャー ID が *8wekyb3d8bbwe*、アプリ識別子が *App* であれば、実行可能ファイルは *Microsoft.WindowsCalculator_8wekyb3d8bbwe!App* になります。

実行可能ファイルパターン UWP アプリの実行可能ファイルパターンは実行可能ファイルとほとんど同じです。たとえば、UWP アプリの実行可能ファイルパターンは、*Microsoft.WindowsCalculator_8wekyb3d8bbwe!App* や **¥Microsoft.WindowsCalculator_8wekyb3d8bbwe!App* になります。

UWP アプリの個々のウィンドウが他の UWP アプリになっていることがあります。このようなウィンドウをテストするには、アプリケーション構成を追加して設定する必要があります。Silk Test Workbench が UWP アプリのウィンドウを解決できない場合、ウィンドウを開いている状態で **アプリケーションの選択** ダイアログ ボックスを開き、**Windows** タブに解決されないウィンドウのエントリが表示されていることを確認します。そのウィンドウに対するアプリケーション構成を追加します。

UWP アプリのテスト時のトラブルシューティング

Silk Test Workbench が UWP アプリのウィンドウを解決できない理由

UWP アプリの個々のウィンドウが他の UWP アプリになっていることがあります。このようなウィンドウをテストするには、アプリケーション構成を追加して設定する必要があります。Silk Test Workbench が UWP アプリのウィンドウを解決できない場合、ウィンドウを開いている状態で **アプリケーションの選択** ダイアログ ボックスを開き、**Windows** タブに解決されないウィンドウのエントリが表示されていることを確認します。そのウィンドウに対するアプリケーション構成を追加します。

Windows API ベースのアプリケーションのサポート

Silk Test Workbench は、Microsoft Windows API ベースのアプリケーションのテストを組み込みでサポートしています。アクセシビリティを有効にすると Microsoft のアプリケーションのいくつかのオブジェクトが Silk Test Workbench によってより詳細に認識されます。たとえば、アクセシビリティを有効にしないと、Silk Test Workbench は Microsoft Word のメニューバーおよびバージョン 7.0 より後の Internet Explorer に表示されるタブについて基本的な情報のみを記録します。ただし、アクセシビリティを有効にすると、Silk Test Workbench によってそれらのオブジェクトがすべて認識されます。必要な場合、新しいウィンドウを定義すると、Silk Test Workbench によるオブジェクトの認識を向上させることもできます。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

サポートするコントロール

Windows ベースのテストで利用可能な記録および再生コントロールの完全な一覧については、「[Win32 クラスリファレンス](#)」を参照してください。

Win32 クラス リファレンス


Win32 アプリケーションを設定すると、Silk Test Workbench は標準の Windows API ベースのコントロールのテストのサポートを組み込みで提供します。

Windows API ベースのクライアント/サーバー アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Windows API ベースのクライアント/サーバー アプリケーションがサポートする属性は次のとおりです。

- caption
- windowid
- priorlabel : 隣接するラベル フィールドのテキストによってテキスト入力フィールドを識別します。通常、フォームのすべての入力フィールドに、入力の目的を説明するラベルがあります。caption のないコントロールの場合、自動的に属性 **priorlabel** がロケーターに使用されます。コントロールの **priorlabel** 値 (テキスト ボックスなど) には、コントロールの左側または上にある最も近いラベルの caption が使用されます。

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

Win32 テクノロジー ドメインにおける priorLabel の決定方法

Win32 テクノロジー ドメインにおいて priorLabel を決定する場合、同じウィンドウ内のすべてのラベルとグループが対象のコントロールとみなされます。以下の条件に従って、コントロールが決定されます。

- コントロールの上または左側にあるラベル、およびコントロールを囲むグループが priorLabel の候補とみなされます。
- 最も単純なケースでは、コントロールに最も近いラベルが priorLabel として使用されます。
- コントロールからの距離が等しい 2 つのラベルが存在する場合、次の条件に基づいて priorLabel が決定されます。
 - 一方のラベルがコントロールの左側にあり、他方が上にある場合、左側のものが優先されます。
 - 両方のラベルがコントロールの左側にある場合、上にあるものが優先されます。
 - 両方のラベルがコントロールの上にある場合、左側のものが優先されます。
- 最も近いコントロールがグループ コントロールである場合、まずグループ内のすべてのラベルが上記の規則に従って決定されます。グループ内に適切なラベルが見つからない場合は、グループのキャプションが priorLabel として使用されます。

埋め込み Chrome アプリケーションのテスト

埋め込み Chrome アプリケーションは、Chromium コアをベースとした埋め込み Web ブラウザー エンジンを使用したデスクトップ アプリケーションです。このようなアプリケーションでは、デスクトップ アプリケーションに Web ブラウザーの機能を追加できます。Chromium Embedded Framework (CEF) や Electron フレームワークを使用して、このようなアプリケーションを作成できます。

Silk Test Workbench は、--remote-debugging-port コマンド ライン引数を指定してリモート デバッグを有効にした埋め込み Chrome アプリケーションのテストをサポートします。Silk Test Workbench で


は、Java (Java AWT や Swing アプリケーションなど) をベースにした埋め込み Chrome アプリケーションのテストをサポートしません。


Silk Test Workbench を使用して埋め込み Chrome アプリケーションをテストするには、アプリケーションの実行可能ファイルにデバッグ ポートを設定する必要があります。コマンド ラインからアプリケーションを開始して、リモート デバッグ ポートを設定します。

- Silk Test Workbench は、`-remote-debugging-port` 引数が埋め込み Chrome アプリケーションのコマンド ライン引数に設定されているかどうかを確認します。引数が設定されている場合、Silk Test Workbench は **埋め込み Chrome サポートの有効化** フィールドに適切な実行可能ファイルとデバッグ用ポートを自動的に設定します。
- `-remote-debugging-port` 引数が埋め込み Chrome アプリケーションのコマンド ライン引数に設定されていない場合は、実行可能ファイルとポートを **埋め込み Chrome サポートの有効化** フィールドに次の手順で指定する必要があります。

1. Silk Test Workbench UI で、**オプションの編集** を選択します。
2. **オプション** ダイアログで、**詳細設定** タブを選択します。
3. **埋め込み Chrome サポートの有効化** オプションで、実行可能ファイルとポートの値ペアを、カンマ区切りで次のように指定します。

```
<application name>.exe=<port number>
```

 **注:** Silk Test Workbench によるリモート デバッグを許可しない埋め込み Chrome アプリケーションをテストすることはできません。

 **注:** Silk Test Workbench では、Electron アプリの非ブラウザ メニューのテストをサポートしません。

例

たとえば、コマンド ラインから *myApp* アプリケーションを次のように実行します。

```
myApp.exe --remote-debugging-port=9222
```

この場合、**埋め込み Chrome サポートの有効化** オプションに実行可能ファイルとポートの値ペアを次のように指定します。

```
myApp.exe=9222
```

MFC (Microsoft Foundation Class) のサポート

MFC (Microsoft Foundation Class) のクラス ID は、時間がたつと変更されることがあるため、安定したロケータを生成するために使用することはできません。不安定なロケータの生成を避けるため、Silk Test Workbench では次の属性をロケータとして使用します。

- MFC クラス名 (MFC コントロールの Windows クラス名が `Afx:` で始まる場合)
- Windows クラス名 (MFC コントロールの Windows クラス名が `Afx:` で始まらない場合)

Silk Test Workbench は MFC バージョン 140 のみをサポートします。また、サポートする組み合わせは次の通りです。

- Release、x86、MBCS
- Release、x86、Unicode
- Debug、x86、MBCS
- Debug、x86、Unicode
- Release、x64、MBCS
- Release、x64、Unicode
- Debug、x64、MBCS
- Debug、x64、Unicode



注: Silk Test Workbench 18.5 以前で生成した MFC コントロールのロケータを持つ既存のテストを実行する場合は、影響するテスト スクリプトで OPT_COMPATIBILITY オプションをバージョン 18.5.0 以前に設定してください。

```
'VB .NET code  
Agent.SetOption("OPT_COMPATIBILITY", "18.5.0")
```

クロスブラウザテスト

Silk Test Workbench を使用すると、単一のポータブルなビジュアル テストやテスト スクリプトでさまざまなブラウザに対して非常に高度な Web アプリケーションの機能を簡単に検証できます。Silk Test Workbench は、最新の Web テクノロジーを使用した、効率的でメンテナンスしやすいクロスブラウザ テストのトップレベルのサポートを提供します。

テスト自動化における主要な困難のひとつが、テストの作成と保守にかかるコストの効率化です。ブラウザによって動作が異なるため、Web アプリケーションの検証を生産的に実行することは困難です。Silk Test Workbench は、クロスブラウザテストにおける次の 3 つの領域をうまく扱うため、ユーザーはテストの記述に集中することができます。

組み込みの同期処理 これにより、サポートするすべてのブラウザ上で実行するスクリプトを作成することができます。AJAX や HTML5 などの高度な動的 Web アプリケーションに特有の非同期イベントを手動で同期する必要はありません。Silk Test Workbench は、HTML や AJAX だけでなく、Apache Flex、Microsoft Silverlight、HTML5/AJAX など、すべての主要な Web 環境に対する同期モードをサポートします。詳細については、「[xBrowser のページ同期](#)」を参照してください。

統合オブジェクトモデル Silk Test Workbench では、さまざまなブラウザ上で幅広く実行するテストを作成、保守できます。すべてのブラウザに対応する統合オブジェクト モデルにより、ユーザーはテストの作成と保守を単一のブラウザを中心に行うことができます。Silk Test Workbench は、すべての他のブラウザ上のオブジェクトに同じ方法でアクセスできるようにすることで、さまざまなブラウザに対する回避策を探ることなく、時間を節約してテストの作成に集中できます。

クロスブラウザスクリプトの記録 スクリプトを記録すると、修正することなくすべての他のブラウザで再生できます。これにより、テスト スクリプトの作成と保守にかかる時間と労力は劇的に減少します。シミュレーションではなく、テストは実際のブラウザ上で実行されます。つまり、テストは、エンドユーザーの動作とまったく同じように動作します。

Silk Test Workbench では、次のブラウザを使用して Web アプリケーションのテストを再生できます。

- Internet Explorer
- Mozilla Firefox (Microsoft Windows 上または macOS 上)
- Google Chrome (Microsoft Windows 上または macOS 上)
- Microsoft Edge
- Chrome for Android (Android デバイス上)
- Apple Safari (macOS 上または iOS デバイス上)
- 埋め込みブラウザ コントロール




注: 次のブラウザのいずれかを使用して Web アプリケーションのテストを記録できます。

- Internet Explorer
- Microsoft Edge
- Mozilla Firefox (Microsoft Windows 上または macOS 上)
- Google Chrome (Microsoft Windows 上または macOS 上)
- モバイル ブラウザ (モバイル デバイス上)

クロス ブラウザー テスト用にスクリプトを記録する場合、Google Chrome、Mozilla Firefox、または Microsoft Edge を使用することを Micro Focus では推奨しています。Internet Explorer を使っ

て Silk Test Workbench で記録したスクリプトは、ほかのブラウザで記録したスクリプトと若干異なる場合があります。

-  **注:** Web アプリケーションを記録または再生する前に、システムにインストールされているすべてのブラウザアドオンを無効にします。Internet Explorer でアドオンを無効にするには、**ツール > インターネット オプション** をクリックし、**プログラム** タブをクリックし、**アドオンの管理** をクリックし、アドオンを選択してから **無効にする** をクリックします。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

サンプル アプリケーション


Silk Test のサンプル Web アプリケーションには、以下の URL からアクセスします。

- <http://demo.borland.com/InsuranceWebExtJS/>
- <http://demo.borland.com/gmopost>
- <http://demo.borland.com/gmoajax>

テストを再生するブラウザの選択

テストを再生するために使用するブラウザを定義できます。

- Silk Test Workbench の UI からテストを実行する場合、**ブラウザの選択** ダイアログ ボックスが表示され、このダイアログ ボックスで選択したブラウザが使用され、テスト スクリプトで設定されているブラウザを Silk Test Workbench は無視します。
- **ブラウザの選択** ダイアログ ボックスが無効の場合 (**再び表示しない** をチェックした場合)、個々のテスト スクリプトのアプリケーション構成によってテストを実行するために使用するブラウザが決定されます。

-  **注:** **ブラウザの選択** ダイアログ ボックスを再び有効にするには、**ツール > オプション** をクリックして、**ブラウザ構成ダイアログの表示** オプションを **はい** に設定します。

- スクリプトをコマンド ラインや CI サーバーから実行する場合は、スクリプトのアプリケーション構成で接続文字列を指定します。

アプリケーション構成で指定したブラウザは、`silktest.configurationName` 環境変数、または `browser` コマンドライン パラメータ を使用して上書き指定できます。

- Silk Central からテストを実行する場合は、テストするブラウザそれぞれの構成を持つ構成スイートを作成します。そして、適切な構成名を指定します。詳細については、『[Silk Central ヘルプ](#)』を参照してください。

silktest.configurationName 環境変数を使用したブラウザの設定の例

- ブラウザーとして Internet Explorer を使用する場合は、次のように入力します。

```
SET silktest.configurationName=InternetExplorer
STW.exe -script MyScript
```



注: Internet Explorer

- ブラウザーとして Microsoft Edge を使用する場合は、次のように入力します。

```
SET silktest.configurationName=Edge
STW.exe -script MyScript
```

- ブラウザーとして Mozilla Firefox を使用する場合は、次のように入力します。

```
SET silktest.configurationName=Firefox
STW.exe -script MyScript
```

- ブラウザーとして Google Chrome を使用する場合は、次のように入力します。

```
SET silktest.configurationName=GoogleChrome
STW.exe -script MyScript
```

- ブラウザーとして Mac 上の Apple Safari を使用する場合は、次のように入力します。

```
SET silktest.configurationName=host=10.0.0.1 - Safari
STW.exe -script MyScript
```

この例では、*host* は Apple Safari をテストする Mac です。ホストは、リモートロケーションとして Silk Test Workbench がインストールされているマシンに接続されている必要があります。詳細については、「リモートロケーションの編集」を参照してください。

- ブラウザーとして Android デバイス上の Google Chrome を使用する場合は、次のような接続文字列を使用します。たとえば、デバイス ID が 11111111 で、デバイスが IP アドレス 10.0.0.1 のリモートマシンに接続されている場合は、次のように入力します。

```
SET
silktest.configurationName="platformName=Android;deviceName=MotoG3;
deviceId=11111111;host=10.0.0.1 - Chrome"
STW.exe -script MyScript
```

- Android デバイス上で Android AOSP Stock Browser を使用する場合は、次のような接続文字列を使用します。たとえば、デバイス ID が 11111111 で、デバイスが IP アドレス 10.0.0.1 のリモートマシンに接続されている場合は、次のように入力します。

```
SET
silktest.configurationName="platformName=Android;deviceName=MotoG3;
deviceId=11111111;host=10.0.0.1 - AndroidBrowser"
STW.exe -script MyScript
```

- ブラウザーとして iOS デバイス上の Apple Safari を使用する場合は、次のような接続文字列を使用します。たとえば、デバイス ID が 11111111 で、デバイスが IP アドレス 10.0.0.1 のリモートマシンに接続されている場合は、次のように入力します。

```
SET silktest.configurationName="platformName=iOS;deviceName=iPad
mini;deviceId=11111111;host=10.0.0.1"
STW.exe -script MyScript
```

さらに、アプリケーション構成でブラウザーを指定する必要があります。



ヒント: すべての例で、環境変数 *silktest.configurationName* を設定する代わりに、Java システムプロパティ - *Dsilktest.configurationName* を設定して、ブラウザーを設定することもできます。たとえば、ブラウザーとして Mac 上の Apple Safari を使用する場合は、次のように入力することもできます。

```
-Dsilktest.configurationName=host=10.0.0.1 - Safari
STW.exe -script MyScript
```



ヒント: Silk Test Workbench の UI から再生または記録を開始すると、**ブラウザーの選択** ダイアログボックスが開き、システムで現在利用可能なブラウザーのリストが表示されます。

browser パラメータを使用したブラウザーの設定の例

- ブラウザーとして Internet Explorer を使用する場合は、次のように入力します。

```
STW.exe -browser InternetExplorer -script MyScript
```

- ブラウザーとして Microsoft Edge を使用する場合は、次のように入力します。

```
STW.exe -browser Edge -script MyScript
```

- ブラウザーとして Mozilla Firefox を使用する場合は、次のように入力します。
STW.exe -browser Firefox -script MyScript
 - ブラウザーとして Google Chrome を使用する場合は、次のように入力します。
STW.exe -browser GoogleChrome -script MyScript
 - ブラウザーとして Mac 上の Apple Safari を使用する場合は、次のように入力します。
STW.exe -connectionstring host=10.0.0.1 - Safari -script MyScript
- この例では、*host* は Apple Safari をテストする Mac です。ホストは、リモートロケーションとして Silk Test Workbench がインストールされているマシンに接続されている必要があります。詳細については、「リモートロケーションの編集」を参照してください。
- ブラウザーとして Android デバイス上の Google Chrome を使用する場合は、次のような接続文字列を使用します。たとえば、デバイス ID が 11111111 で、デバイスが IP アドレス 10.0.0.1 のリモートマシンに接続されている場合は、次のように入力します。
STW.exe -connectionstring "platformName=Android;deviceName=MotoG3;host=10.0.0.1 - Chrome" -script MyScript
 - Android デバイス上で Android AOSP Stock Browser を使用する場合は、次のような接続文字列を使用します。たとえば、デバイス ID が 11111111 で、デバイスが IP アドレス 10.0.0.1 のリモートマシンに接続されている場合は、次のように入力します。
STW.exe -connectionstring "platformName=Android;deviceName=MotoG3;host=10.0.0.1 - AndroidBrowser" -script MyScript
 - ブラウザーとして iOS デバイス上の Apple Safari を使用する場合は、次のような接続文字列を使用します。たとえば、デバイス ID が 11111111 で、デバイスが IP アドレス 10.0.0.1 のリモートマシンに接続されている場合は、次のように入力します。
STW.exe -connectionstring "platformName=iOS;deviceName=iPad mini;host=10.0.0.1" -script MyScript
- さらに、アプリケーション構成でブラウザーを指定する必要があります。

xBrowser でのテストオブジェクト

Silk Test Workbench では、以下のクラスを使用して Web アプリケーションがモデル化されます。

クラス	説明
BrowserApplication	Web ブラウザのメイン ウィンドウを公開し、タブ化するための方法を提供します。
BrowserObject	BrowserApplication に含まれるすべてのオブジェクトの基底クラスを表します。
BrowserWindow	タブおよび埋め込みブラウザ コントロールへのアクセスを提供し、異なるページに移動するための方法を提供します。
DomElement	Web アプリケーションの DOM ツリー (フレームを含む) を提供し、すべての DOM 属性へのアクセスを提供します。一部の DOM 要素では、特殊なクラスを使用できます。

クラス	説明
DomButton	Web ページの最上位のコンテナを表します。 DomElement を介して DOM ツリーを公開します。
DomCheckBox	<input type='checkbox'> タグを使用して指定された、すべての DOM 要素を表します。
DomForm	<form> タグを使用して指定された、すべての DOM 要素を表します。
DomLink	<a> タグを使用して指定された、すべての DOM 要素を表します。
DomListBox	<select> タグを使用して指定された、すべての DOM 要素を表します。
DomRadioButton	<input type='radio'> タグを使用して指定された、すべての DOM 要素を表します。
DomTable	<table> タグを使用して指定された、すべての DOM 要素を表します。
DomTableRow	<tr> タグを使用して指定された、すべての DOM 要素を表します。
DomTextField	次のタグのいずれかを使用して指定された、すべての DOM 要素を表します。 <ul style="list-style-type: none"> • <input type='text'> • <input type='password'> • <input type='file'> • <textarea>

xBrowser オブジェクトに対するオブジェクト解決

xBrowser テクノロジ ドメインは動的オブジェクト解決をサポートします。

テストケースは、オブジェクトを検索し、識別するためにロケータ文字列を使用します。一般的なロケータには、`"//LocatorName[@locatorAttribute='value']"` のようにロケータ名と少なくとも 1 つのロケータ属性が含まれます。

ロケータ名 Java SWT などの他の種類のテクノロジーでは、テスト オブジェクトのクラス名を使用してロケータ名が作成されます。xBrowser では、DOM 要素のタグ名もロケータ名として使用できます。以下のロケータは、同じ要素を示しています。

1. タグ名を使用した場合：`"//a[@href='http://www.microfocus.com']"`
2. クラス名を使用した場合：`"//DomLink[@href='http://www.microfocus.com']"`


再生速度を最適化するには、クラス名ではなくタグ名を使用します。

ロケータ属性 すべての DOM 属性は、ロケータ文字列属性として使用できます。たとえば、要素 `<button automationid='123'>Click Me</button>` はロケータ `"//button[@automationid='123']"` を使用して識別できます。

ロケータの記録 Silk Test Workbench では、テストを記録したり、**オブジェクトの識別** ダイアログ ボックスを使用したりするときに、組み込みロケータ生成プログラムが使用されます。特定のアプリケーションの結果を向上するように、ロケータ生成プログラムを構成することができます。

xBrowser のページ同期

同期は、すべてのメソッド呼び出しの前後に自動的に実行されます。メソッド呼び出しは、同期条件が満たされるまで開始せず、終了もしません。

 **注:** プロパティのアクセスは同期されません。

同期モード

Silk Test Workbench には、HTML および AJAX 用の同期モードがあります。


HTML モードを使用すると、すべての HTML ドキュメントが対話的な状態になることが保証されます。このモードでは、単純な Web ページをテストすることができます。Java Script が含まれるより複雑なシナリオが使用される場合は、以下の同期関数を使用して、手動でスクリプトを記述することが必要になることがあります。

- WaitForObject
- WaitForProperty
- WaitForPropertyNotEquals
- WaitForDisappearance
- WaitForChildDisappearance
- WaitForScreenshotStable

AJAX モードでは、ブラウザがアイドル状態に類似した状態になるまで待機します。このことは、AJAX アプリケーションまたは AJAX コンポーネントを含むページに対して特に効果的です。AJAX モードを使用すると、同期関数を手動で記述する必要がなくなるため、スクリプト（オブジェクトの表示または非表示を待機したり、特定のプロパティ値を待機するなど）の作成処理が大幅に簡略化されます。また、この自動同期は、スクリプトを手動で適用しないで記録と再生を正常に行うための基礎となります。

トラブルシューティング


AJAX の非同期の特性のため、ブラウザが完全にアイドル状態になることはありません。このため、Silk Test Workbench でメソッド呼び出しの終了が認識されず、特定のタイムアウト時間が経過したあとで、タイムアウトエラーが発生することがまれにあります。この場合は、少なくとも、問題が発生する呼び出しに対して、同期モードを HTML に設定する必要があります。

 **注:** 使用するページ同期メソッドにかかわらず、Flash オブジェクトがサーバーからデータを取得し、計算を実行してデータをレンダリングするテストでは、手動でテストに同期メソッドを追加する必要があります。メソッドを追加しないと、Silk Test Workbench は、Flash オブジェクトが計算を完了するまで待機しません。たとえば、ビジュアルテストでは Delay を使用し、スクリプトでは System.Threading.Thread.Sleep(milliseconds) を使用します。

AJAX フレームワークやブラウザによっては、サーバーから非同期にデータを取得するために、特殊な HTTP 要求を継続して出し続けるものがあります。これらの要求により、指定した同期タイムアウトの期限が切れるまで同期がハングすることがあります。この状態を回避するには、HTML 同期モードを使用するか、問題が発生する要求の URL を **同期除外リスト** 設定で指定します。

URL を除外フィルタに追加するには、**オプション** ダイアログ ボックスの **同期除外リスト** で URL を指定します。

監視ツールを使用して、同期の問題により再生エラーが発生するかどうかを判断します。たとえば、FindBugs (<http://findbugs.sourceforge.net/>) を使用して、AJAX 呼び出しが再生に影響を及ぼしているかどうかを判断できます。次に、問題が発生するサービスを **同期除外リスト** に追加します。

 **注:** URL を除外すると、指定した URL を対象とする各呼び出しに対して同期が無効になります。その URL に対して必要な同期は、手動で呼び出す必要があります。たとえば、WaitForObject をスクリプトまたはビジュアルテストに手動で追加する必要がある場合があります。手動で数多くの呼び出し

を追加することを避けるために、可能なかぎり、最上位の URL ではなく、具体的に対象を絞って URL を除外します。

ページ同期設定の構成

オプション ダイアログ ボックスでは、各ビジュアル テストまたはスクリプトのページ同期設定を個別に構成したり、すべてのテストおよびスクリプトに適用するグローバル オプションを設定したりできます。これらのオプションを構成するには、**ツール > オプション** を選択し、**再生** を展開し、**xBrowser** をクリックします。

ビジュアル テストの個別の設定を構成するには、ビジュアル テストを記録し、次に、グローバル再生値を上書きする **再生設定の設定** ステップを挿入します。設定には、**同期除外リスト**、**同期モード**、**同期タイムアウト** があります。

たとえば、ビジュアル テストでは、**同期モード** 再生設定を **HTML** に設定し、残りのステップでは必要に応じて **同期モード** を **AJAX** に戻します。

スクリプトの個別の設定を構成するには、以下のいずれか呼び出します。

- `Agent.SetOption(Options.XBrowserSynchronizationMode, mode)`

mode には、HTML を使用する場合は 1、AJAX を使用する場合は 2 を入力します。

- `Agent.SetOption(Options.XBrowserSynchronizationExcludes, URL)`

除外するサービスまたは Web ページの URL または URL の一部を指定します。たとえば、`http://example.com/syncsample/timeService` と入力します。

- `Agent.SetOption(Options.XBrowserSynchronizationTimeout, time)`

time をミリ秒で指定します。

xBrowser における API 再生とネイティブ再生の比較

Silk Test Workbench では、Web アプリケーション用に API 再生とネイティブ再生がサポートされています。アプリケーションでプラグインまたは AJAX を使用している場合は、ユーザーの入力そのものを使用します。アプリケーションでプラグインまたは AJAX を使用していない場合は、API 再生を使用することをお勧めします。

ネイティブ再生には以下のような利点があります。

- ネイティブ再生では、マウス ポインタを要素上に移動し、対応する要素を押すことによって、エージェントはユーザー入力をエミュレートします。この結果、再生はほとんどのアプリケーションで変更なしで動作します。
- ネイティブ再生では、Flash や Java アプレットなどのプラグイン、および AJAX を使用するアプリケーションをサポートしていますが、高レベルの API 記録はサポートしていません。

API 再生には以下のような利点があります。

- API 再生では、Web ページが `onmouseover` や `onclick` などの DOM イベントによって直接実行されます。
- API 再生を使用するスクリプトでは、ブラウザをフォアグラウンドで実行する必要はありません。
- API 再生を使用するスクリプトでは、要素をクリックする前に、要素が表示されるようにスクロールする必要はありません。
- 一般的に、高レベルのユーザー入力は再生中にポップアップ ウィンドウやユーザー対話の影響を受けないため、API スクリプトの信頼性は高くなります。
- API 再生は、ネイティブ再生よりも高速です。

オプション ダイアログ ボックスを使用して、記録する関数の種類を構成したり、ユーザーの入力そのものを使用するかどうかを指定したりできます。

API 再生とネイティブ再生の関数の違い

DomElement クラスには、API 再生とネイティブ再生に対して異なる関数が備えられています。

以下の表に、API 再生とネイティブ再生で使用する関数を示します。

	API 再生	ネイティブ再生
マウス操作	DomClick	Click
	DomDoubleClick	DoubleClick
	DomMouseMove	MoveMouse
		PressMouse
		ReleaseMouse
キーボード操作	使用不可	TypeKeys
特殊な関数	Select	使用不可
	SetText	
	など	

マウス移動の詳細設定

マウス移動イベントを使用する Web アプリケーション、Win32 アプリケーション、および Windows Forms アプリケーションでマウス移動操作を記録するかどうかを指定します。たとえば、Apache Flex や Swing など、xBrowser テクノロジ ドメインの子ドメインのマウス移動イベントを記録することはできません。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプション が右側のパネルに表示されます。
3. **全般** をクリックします。
4. マウス移動操作を記録するには、**マウスの移動操作を記録する** リスト ボックスから **はい** を選択します。マウスの移動イベントを使用する Web ページをテストする場合は、**はい** を選択します。
Silk Test Workbench では、スクリプトを短くするために、マウスが置かれた要素またはその親が変化するマウスの移動イベントのみが記録されます。
5. マウスの移動操作を記録する場合、MoveMouse 操作が記録される前に、どのくらいの間マウスが不動状態になければならないかを、**マウスの移動記録遅延** テキスト ボックスにミリ秒単位で指定します。デフォルト値は、200 に設定されています。
マウスの移動操作は、この時間、マウスが静止している場合にのみ記録されます。遅延を短くすると、予期しないマウスの移動操作が増加します。遅延を長くすると、操作を記録するためにマウスを静止しておく必要があります。
6. **OK** をクリックします。

xBrowser のブラウザ構成の設定

いくつかのブラウザ設定は、テストを継続的に安定して実行するのに役立ちます。設定を変更しなくても Silk Test Workbench は動作しますが、ブラウザ設定を変更するにはいくつかの理由があります。

再生速度を向上させる 読み込みに時間を要する Web ページではなく、about:blank をホーム ページとして使用する

ブラウザの予期しない動作を回避する

- ポップアップ ウィンドウや警告ダイアログ ボックスを無効にする
- オート コンプリート機能を無効にする
- パスワード ウィザードを無効にする

ブラウザの誤動作を防止する

不要なサードパーティ製プラグインを無効にする

以下のセクションでは、対応するブラウザにおけるこれらの設定場所について説明します。

Internet Explorer

ブラウザ設定は、**ツール > インターネット オプション** にあります。以下の表に、調整できるオプションの一覧を示します。

タブ	オプション	設定	コメント
全般	ホーム ページ	about:blank に設定します。	新しいタブの起動時間を最小限に抑えます。
全般	タブ	<ul style="list-style-type: none"> • 複数のタブを閉じるときの警告を無効にします。 • 新しいタブを作成したとき、新しいタブに切り替えます。 	<ul style="list-style-type: none"> • 予期しないダイアログ ボックスが表示されないようにします。 • このようにしないと、新しいタブを開くリンクが正しく再生されない場合があります。
プライバシー	ポップアップ ブロック	ポップアップ ブロックを無効にします。	Web サイトで新しいウィンドウを開くことができることを確認します。
コンテンツ	オートコンプリート	完全にオフにします。	<ul style="list-style-type: none"> • 予期しないダイアログ ボックスが表示されないようにします。 • キー入力するときに予期しない動作を回避します。
プログラム	アドオンの管理	最低限必要なアドオンのみを有効にします。	<ul style="list-style-type: none"> • サードパーティ製アドオンにはバグが含まれていることがあります。 • Silk Test Workbench と互換性がない可能性があります。
詳細設定	設定	<ul style="list-style-type: none"> • Internet Explorer の更新について自動的に確認する を無効にします。 • スクリプトのデバッグを使用しない (Internet Explorer) を有効にします。 • スクリプトのデバッグを使用しない (その他) を有効にします。 • 自動クラッシュ回復機能を有効にする を無効にします。 • スクリプト エラーごとに通知を表示する を無効にします。 • すべての ...警告する 設定を無効にします。 	予期しないダイアログ ボックスが表示されないようにします。



注: 100% 以外の拡大レベルを使用して Internet Explorer で Web アプリケーションを記録すると、期待通り機能しない可能性があります。Internet Explorer で Web アプリケーションに対する操作を記録する前に、拡大レベルを 100% に設定してください。


Mozilla Firefox

Mozilla Firefox のブラウザ設定を変更する必要はありません。Silk Test Workbench により、適切なコマンドラインパラメータが指定され、自動的に Mozilla Firefox が起動します。

 **注:** Web アプリケーションのテスト時に予期しない動作を避けるため、Mozilla Firefox の自動更新を無効にします。詳細については、『[Stop automatic updates](#)』を参照してください。

Google Chrome

Google Chrome のブラウザ設定を変更する必要はありません。Silk Test Workbench により、適切なコマンドラインパラメータが指定され、自動的に Google Chrome が起動します。

 **注:** Web アプリケーションのテスト時に予期しない動作を避けるため、Google Chrome の自動更新を無効にします。詳細については、『[Turning Off Auto Updates in Google Chrome](#)』を参照してください。

リモート デスクトップ ブラウザーの接続文字列

接続文字列は、テストに使用するリモート デスクトップ ブラウザーを指定します。リモート ブラウザーで Web アプリケーションをテストする場合、Silk Test Workbench はリモート ロケーションに接続するために接続文字列を使用します。接続文字列は、アプリケーション構成の主要な一部です。テストする Web アプリケーションを構成するときに、接続文字列は設定されます。接続文字列を変更するには、**アプリケーション構成の編集** ダイアログ ボックスを使用します。

リモート ブラウザーで Web アプリケーションをテストする場合、接続文字列にはリモート マシンの IP アドレスまたはホスト名を意味する host だけが含まれます (10.0.0.1 など)。正しいブラウザを選択するために、Silk Test Workbench はブラウザの種類と共に接続文字列を使用します。ブラウザの種類は、**アプリケーション構成の編集** ダイアログ ボックスでも指定できます。

ホスト名は、大文字と小文字は区別されません。

 **注:** リモート デスクトップ ブラウザーのテストは、リモート Microsoft Windows マシン上の Microsoft Edge およびリモート Mac 上の Apple Safari に対してのみサポートされます。

接続文字列の例

```
"host=10.0.0.1"
```

リモート Windows マシン上でのブラウザのテスト

Microsoft Edge、Google Chrome、Mozilla Firefox をリモート Windows マシン上でテストするには、リモート マシン上に Silk Test Workbench をインストールする必要があります。Microsoft Edge は、Microsoft Windows 10 マシン上でのみ実行できます。Silk Test Workbench は、リモート Windows マシン上の Mozilla Firefox 55 以前のテストをサポートします。


1. ブラウザーのテストを実行するローカル マシン上で、リモート Windows マシンをリモート ロケーションとして追加します。
詳細については、「[リモート ロケーションの編集](#)」を参照してください。
2. Silk Test Information Service がすでにリモート マシン上で管理者権限で実行されている場合は、Silk Test Information Service を無効化します。
 - a) リモート マシンに管理者としてログインします。
 - b) **コントロール パネル** (アイコン表示) を開きます。
 - c) **管理ツール** をクリックします。
 - d) **サービス** をダブルクリックします。
 - e) Silk Test Information Service をダブルクリックします。

- f) サービスが実行中の場合、**停止** をクリックして、サービスの状態が 停止 になるまで待機します。
- g) **スタートアップの種類** を **無効** に変更します。
- h) **OK** をクリックします。

3. Silk Test Information Service を標準の整合性レベル（管理者権限無し）で開始します。

- a) ファイル エクスプローラーを開き、%OPEN_AGENT_HOME%/InfoService に移動します。
たとえば、C:\Program Files (x86)\Silk\SilkTest\ng\InfoService になります。
- b) infoservice_start.bat をダブルクリックします。

4. 以上で、リモートマシン上のブラウザーを **アプリケーションの選択** ダイアログ ボックスで選択できるようになります。

 **注:** Silk Test Information Service が、サービス セッションではなくユーザー セッションで実行されていることを確認します。Silk Test Information Service は、UI 操作が有効でなければなりません。セッション 0 であるサービス セッションは、UI 操作が有効ではありません。

Mac 上の Google Chrome または Mozilla Firefox のテスト

リモート macOS マシン上の Google Chrome または Mozilla Firefox をテストするには、リモートマシン上に Silk Test Information Service をインストールする必要があります。詳細については、「[Silk Test Information Service を Mac にインストールする](#)」を参照してください。

1. Google Chrome または Mozilla Firefox をテストするローカルマシン上で、リモート macOS マシンをリモート ロケーションとして追加します。
詳細については、「[リモート ロケーションの編集](#)」を参照してください。
2. 以上で、リモート macOS マシン上の Google Chrome または Mozilla Firefox を **アプリケーションの選択** ダイアログ ボックスで選択できるようになります。
3. STW.EXE コマンド ラインを使って、macOS マシン上のリモート ブラウザーに対するスクリプトを実行するには、次のコマンドを使用します：


```
stw -d empty -browser <browsername> -connectionstring <connectionstring> -s <Scriptname>
```

例：

```
stw -d empty -browser Firefox -connectionstring "host=try-imac" -s BrowserScript
```

または

```
stw -d empty -verbose -browser GoogleChrome -connectionstring "host=try-imac" -s BrowserScript
```

 **注:** Silk Test Information Service が、サービス セッションではなくユーザー セッションで実行されていることを確認します。Silk Test Information Service は、UI 操作が有効でなければなりません。セッション 0 であるサービス セッションは、UI 操作が有効ではありません。

Google Chrome 上でテストを実行する際の前提条件と制限事項についての情報は、「[Google Chrome を使用したテスト](#)」セクションにあるトピックを参照してください。Mozilla Firefox 上でテストを実行する際の前提条件と制限事項についての情報は、「[Mozilla Firefox を使用したテスト](#)」セクションにあるトピックを参照してください。

WebDriver ベースのブラウザーのケイパビリティの設定

WebDriver ベースのブラウザー上で Web アプリケーションをテストする場合は、ケイパビリティを設定してブラウザー セッションの設定をカスタマイズできます。

Silk Test Workbench では、次のブラウザーの種類に対する接続文字列で WebDriver ケイパビリティを指定できます。

- Google Chrome
- Mozilla Firefox

Mozilla Firefox 48 以降で利用可能なオプションとケイパビリティについての詳細は、『<https://github.com/mozilla/geckodriver>』を参照してください。Google Chrome で利用可能なオプションとケイパビリティについての詳細は、『[Capabilities & ChromeOptions](#)』を参照してください。

Silk Test Workbench でケイパビリティを設定するには：

1. ケイパビリティを変更する Web アプリケーションのプロジェクトを選択します。
2. プロジェクトの基本状態の接続文字列を編集します。
接続文字列は、次の方法で編集できます。
 - **アプリケーション構成の編集** ダイアログ（カスタマイズしたブラウザーに対する操作を記録する場合など）
 - スクリプト（カスタマイズしたブラウザーに対してテスト スクリプトを実行するだけの場合）
 詳細については、「基本状態」を参照してください。
3. スクリプトを実行して、指定したオプションとケイパビリティでブラウザーを開始します。

例

Mozilla Firefox から実行可能ファイルを自動的にダウンロードするには、スクリプトの基本状態に次のコードを追加します。

```
' VB .NET code - No linebreaks, this is a single line of code.
baseState.ConnectionString = "moz:firefoxOptions={¥"prefs¥":
{ ¥"browser.download.folderList¥": 2,
¥"browser.helperApps.neverAsk.saveToDisk¥": ¥"application/octet-stream
¥"}};"
```

Mozilla Firefox のダウンロード フォルダを指定するには、スクリプトの基本状態に次のコードを追加します。

```
' VB .NET code
baseState.ConnectionString = "moz:firefoxOptions={¥"prefs¥":
{ ¥"browser.download.dir¥" : ¥"C:/Download¥"}};"
```

Mozilla Firefox のコマンド ライン引数を設定するには、スクリプトの基本状態に次のコードを追加します。

```
' VB .NET code
baseState.ConnectionString = "moz:firefoxOptions={¥"args¥": [¥"--devtools
¥"}];"
```

Google Chrome から実行可能ファイルを指定したフォルダに自動的にダウンロードするには、スクリプトの基本状態に次のコードを追加します。

```
' VB .NET code - No linebreaks, this is a single line of code.
baseState.ConnectionString = "chromeOptions={¥"prefs¥":
{¥"profile.default_content_setting_values.automatic_downloads¥":1,
¥"download.default_directory¥":¥"c:/Download¥",
¥"download.prompt_for_download¥":false}};"
```

Google Chrome のパスワード マネージャーにメッセージを表示させないようにするには、スクリプトの基本状態に次のコードを追加します。

```
' VB .NET code - No linebreaks, this is a single line of code.
baseState.ConnectionString = "chromeOptions={¥"args¥": [¥"--disable-save-
password-bubble¥"], ¥"prefs¥": {¥"profile.password_manager_enabled
¥":false, ¥"credentials_enable_service¥":false}};"
```

Mac 上の Apple Safari を使用したテスト

このセクションでは、Silk Test Workbench がインストールされている Windows マシンと接続し、Mac マシン上の Apple Safari をテストすることによって、クロス ブラウザー テスト セットを拡張する方法について説明します。

Mac 上の Apple Safari を使用したテストにおける前提条件

Mac 上の Apple Safari を使用してテストを行う前に、次の前提条件を満たしていることを確認してください。

- Mac がリモート ロケーションとして Silk Test Workbench がインストールされている Windows マシンに接続されている。詳細については、「リモート ロケーションの編集」を参照してください。
- Apple Safari 9 でテストする場合、Apple Safari 用の WebDriver 拡張である SafariDriver が Mac 上にインストールされている (SafariDriver は従来のクライアント/サーバーの関係を逆転し、WebSocket を使用して WebDriver クライアントと通信します)。Apple Safari 10.1 を使用する場合は、Safari はドライバーの実装を組み込みで備えています。
- Java JDK が Mac 上にインストールされている。
- Information Service が Mac 上にインストールされている。Information Service に必要なファイルを取得するには、Silk Test インストーラを使用します。詳細については、「[Silk Test Information Service を Mac にインストールする](#)」を参照してください。
- Apple Safari 上でテストを実行するには、Information Service をインストールしたユーザーが Mac にログインしている必要があります。



ヒント: Micro Focus では、起動時に正しいユーザーで Mac に自動的にログインするように設定することを推奨します。詳細については、「[起動時に自動的にログインするように Mac を設定する](#)」を参照してください。

- Mac 上の Apple Safari に対して無人テストを実行する場合は、**システム環境設定** の **省エネルギー** ペインで、次の設定を行います。
 - **ディスプレイをオフにするまでの時間** を **しない** に設定します。
 - **ディスプレイがオフのときにコンピュータを自動でスリープさせない** チェック ボックスをオンにします。



注: これらの設定は、**Silk Test Configuration Assistant** を使用すると簡単に行えます。Mac 上で **Configuration Assistant** を開くには、ステータス メニューの Silk Test アイコンをクリックして、**Configuration Assistant** を選択します。

- Mac 上の Apple Safari に対して無人テストを実行するには、スクリーンセーバーを無効にします。

1. **システム環境設定 > デスクトップとスクリーンセーバ** を開きます。
2. **スクリーンセーバ** タブをクリックします。
3. **開始までの時間** を **開始しない** に設定します。



注: これらの設定は、**Silk Test Configuration Assistant** を使用すると簡単に行えます。Mac 上で **Configuration Assistant** を開くには、ステータス メニューの Silk Test アイコンをクリックして、**Configuration Assistant** を選択します。

- Apple Safari 10.1 を使用してテストする場合は、Safari 開発メニューを有効にします。**Safari > 環境設定** を選択して、**詳細** をクリックし、**メニューバーに '開発' メニューを表示** をオンにします。
- Apple Safari 10.1 を使用してテストする場合は、リモート オートメーションを有効にします。Safari 開発メニューから、**リモート オートメーションを許可** をオンにします。
- 初めて Apple Safari 10.1 でテストを実行する際に、パスワード入力が必要になる場合があります。

Apple Safari のテストの準備


Apple Safari 10.1 以降で Web アプリケーションをテストする場合、**Silk Test Configuration Assistant** を使うと、簡単に Apple Safari を設定できます。Mac 上で **Configuration Assistant** を開くには、ステータス メニューの Silk Test アイコンをクリックして、**Configuration Assistant** を選択しま

す。または、「[Mac 上の Apple Safari を使用したテストにおける前提条件](#)」に書かれている要件を満たすよう、次の手順を実行します。


1. Apple Safari のリモート オートメーションを有効化するために、**開発** メニューを開き、**リモート オートメーションを許可** をオンにします。
開発 メニューは、デフォルトでは表示されません。このメニューを開くには：
 - a) **Safari** メニューから **環境設定** を選択します。
 - b) **環境設定** ウィンドウで、**詳細** タブを選択します。
 - c) **メニュー バーに "開発" メニューを表示** チェック ボックスをオンにします。
 - d) **環境設定** ウィンドウを閉じます。
2. Apple Safari 上で最初にテストを実行すると、ブラウザー ウィンドウが自動テストによってリモート制御されていることを示すダイアログ ボックスが表示されます。**セッションを続ける** をクリックします。

Apple Safari と Selenium WebDriver についての詳細は、『<https://webkit.org/blog/6900/webdriver-support-in-safari-10/>』を参照してください。

Silk Test Information Service を Mac にインストールする


 **注:** Information Service を Mac にインストールするには、Mac の管理者権限が必要です。

Mac 上の Apple Safari や、iOS デバイスや Android デバイスを使ってテストを作成、実行するには、Mac 上に Silk Test Information Service (Information Service) をインストールします。Information Service をインストールして実行すると、Windows マシン上にインストールした Silk Test Workbench クライアントからテストを記録、再生できるようになります。

 **注:** Mac 上で記録することはできません。Mac をテスト ロケーションとして Silk Test Workbench に追加する場合は、Silk Test Workbench ドキュメントの「リモート ロケーションの編集」を参照してください。

Information Service を Mac にインストールするには：

1. Silk Test Workbench ドキュメント「[Mac 上の Apple Safari を使用したテストにおける前提条件](#)」を参照してください。
2. Java JDK が Mac 上にインストールされていることを確認します。
3. iOS デバイス上でモバイル アプリケーションをテストする場合は、Xcode が Mac 上にインストールされていることを確認します。
4. Information Service セットアップ ファイル (SilkTestInformationService<バージョン>-<ビルド番号>.pkg) にアクセスします。
 - Silk Test のインストール時に Information Service セットアップ ファイルをダウンロードした場合は、Silk Test インストール ディレクトリ (C:\Program Files (x86)\Silk\SilkTest など) の macOS フォルダを開きます。
 - Silk Test のインストール時に Information Service セットアップ ファイルをダウンロードしなかった場合は、[Micro Focus SupportLine](#) からセットアップ ファイルをダウンロードできます。
5. SilkTestInformationService<バージョン>-<ビルド番号>.pkg ファイルを Mac にコピーします。
6. SilkTestInformationService<バージョン>-<ビルド番号>.pkg を実行して、Information Service をインストールします。
7. インストール ウィザードの指示に従います。
8. パスワードを尋ねられた場合、現在サインインしている Mac ユーザーのパスワードを入力します。
9. Apple Safari が開き、SafariDriver を信頼するかどうかを尋ねるメッセージ ボックスが表示されたら、**信頼** をクリックします。

 **注:** macOS 10.12 (Sierra) 以前上で Apple Safari 10 以前に対してテストを実行する場合は、Mac 上に SafariDriver をインストールする必要があります。リモート接続ではなく、直接 Mac にログインしている場合には、SafariDriver だけをインストールできます。


10 インストールを完了するために、現在の Mac ユーザーをログアウトします。Information Service が正しくインストールされていることを確認するには、Mac にログインします。

11 macOS Mojave (10.14) 以降の Mac に Information Service をインストールした場合、Mac にログインした後で Silk Test に対するオートメーションの承認を許可する必要があります。

許可を与える必要がある場合、Silk Test は要求の承認ダイアログを自動的に表示します。

a) **OK** をクリックして、許可して情報ダイアログを閉じます。


b) その後に表示されるすべての要求の承認ダイアログに対して、**OK** をクリックします。

 **重要:** これらの Silk Test に対する承認を許可しないと、Google Chrome 上の Web アプリケーションや iOS デバイスやシミュレーター上のモバイル アプリケーションを、この Mac でテストできなくなります。いずれかの承認ダイアログで、誤って **許可しない** をクリックしてしまった場合は、Mac 上でターミナルを開き、次のコマンドを実行してください。

```
sudo tccutil reset AppleEvents
```

その後、Mac を再起動して、再度承認ダイアログが表示されたら **OK** をクリックして許可します。

12 画面の右上にある Silk Test アイコンをクリックすると、利用可能なデバイスをブラウザーを確認できます。

 **ヒント:** Silk Test アイコンが表示されない場合は、Mac を再起動してください。

Apple Safari を使用したテストの制限事項

以下に、Mac 上の Apple Safari を使用してテストする際の既知の制限事項を一覧します。

- 次のクラス、インターフェイス、メソッド、プロパティは、Mac 上の Apple Safari を使用した Web アプリケーションのテストでは現時点ではサポートされません：
 - BrowserApplication クラス。
 - ClearCache メソッド
 - CloseOtherTabs メソッド
 - CloseTab メソッド
 - ExistsTab メソッド
 - GetHorizontalScrollbar メソッド
 - GetNextCloseWindow メソッド
 - GetSelectedTab メソッド
 - GetSelectedTabIndex メソッド
 - GetSelectedTabName メソッド
 - GetTabCount メソッド
 - GetVerticalScrollbar メソッド
 - IsActive メソッド
 - Minimize メソッド
 - OpenContextMenu メソッド
 - OpenTab メソッド
 - Restore メソッド
 - SelectTab メソッド
 - SetActive メソッド
 - WindowState プロパティ
 - BrowserWindow クラス。
 - AcceptAlert メソッド

- DismissAlert メソッド
- GetAlertText メソッド
- IsAlertPresent メソッド
- MouseMove メソッド
- PressKeys メソッド
- PressMouse メソッド
- ReleaseKeys メソッド
- ReleaseMouse メソッド
- IMoveable クラス。
 - GetFocus メソッド。
- Silk Test Workbench は、TypeKeys メソッドで **CMD** キーをサポートしません。
- Silk Test Workbench は、Apache Flex のテストをサポートしません。
- Silk Test Workbench は、Apple Safari 上の JavaScript ソースを使った iframe のテストをサポートしません。
- Apple Safari 上で HTTPS を使用したセキュアな Web アプリケーションをテストするには、必要なすべてのサーバー証明書が信頼済みであることを確認してください。
- Silk Test Workbench は、Apple Safari のネイティブ サポートは提供しません。内部 Apple Safari 機能をテストすることはできません。たとえば、テストで、ナビゲーション バーにテキストを追加して、現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して Web ページ間を移動できます。
- Silk Test Workbench は、Apple Safari に対する JavaScript ダイアログ API 関数をサポートしません。回避策として、このような関数が無視されるようにパッチを当てることができます。詳細については、「<https://groups.google.com/forum/#!topic/selenium-developer-activity/qs0vJw93g9c>」を参照してください。
- Silk Test Workbench は、Apple Safari のタブをサポートしません。
- ビジュアル テストの再生速度が遅い場合は、**ツール > オプション > 再生 > 結果 > ビジュアル テスト** のオプション **コントロールのキャプチャ** を **いいえ** に設定します。そうしないと、リッチな Web ページでは再生速度が著しく低下し、通信がタイムアウトすることもあります。
- 複数ウィンドウのアプリケーションをテストするには、Apple Safari のポップアップ ブロックを解除してください。解除する場合は、Apple Safari を起動し、**Safari 環境設定 > セキュリティ > ポップアップウィンドウを開かない** のチェックを外します。
- Silk Test Workbench は、パスワードを保存するダイアログ ボックスのテストをサポートしません。このダイアログ ボックスを表示しないようにする場合は、Apple Safari を起動し、**Safari 環境設定 > 自動入力** に移動して **ユーザー名とパスワード** チェック ボックスのチェックを外します。
- Silk Test Workbench は、Apple Safari では XPath 式のプロパティをサポートしません。XPath 式では、属性のみがサポートされます。
- Silk Test Workbench は、Content-Security-Policy HTTP ヘッダーを含んだ Web アプリケーションのテストをサポートしません。
- Apple Safari 10.1 を使用する場合、Silk Test Workbench はブラウザーの前に戻る操作をサポートしません。
- Apple Safari 10.1 を使用する場合、Silk Test Workbench は TypeKeys メソッドでの Ctrl キーの使用をサポートしません。
- Apple Safari 10.1 を使用する場合、Silk Test Workbench は Frame および IFrame 内では DOM 操作のみをサポートします。
- Apple Safari 10.1 を使用する場合、Silk Test Workbench は Frame および IFrame での移動操作をサポートしません。
- Apple Safari 10.1 を使用する場合、Silk Test Workbench は記録時の直接スクロールをサポートしません。回避策として、executeJavaScript メソッドを使用できます。

複数の Apple Safari テストの同時実行

Apple Safari 上でテストを実行するには、Silk Test がインストールされた Windows マシンに接続された Mac が必要です。複数の Apple Safari テストを Apple Safari 上で実行する場合、これらのテストは同じ Mac 上で同時に実行されることになります。



注: Mac 上の Apple Safari に対して実行されるテストそれぞれが、Apple Safari のインスタンスを個々に開きます。数多くの Apple Safari のインスタンスが同時に実行すると、Mac のパフォーマンスが低下する可能性があります。

Silk Test Information Service を Mac からアンインストールする

Mac 上の Apple Safari に対するテストを実行する必要がなくなった場合など、次の手順で Silk Test Information Service を Mac からアンインストールすることができます。

1. Mac 上で、`uninstallInfoService.sh` のような新しいシェル ファイルを作成します。
2. 新しいファイルに以下のコードを入力します。

```
#!/bin/sh

if launchctl list | grep com.borland.infoservice ; then
    launchctl unload /Library/LaunchAgents/com.borland.infoservice.plist
    echo "unloading Launch Daemon"
fi

if [ -d "/Applications/Silk" ]
then
    sudo rm -rf /Applications/Silk
fi

if [ -f "/Library/LaunchAgents/com.borland.infoservice.plist" ]
then
    sudo rm /Library/LaunchAgents/com.borland.infoservice.plist
fi

if [ -f "/usr/local/bin/ideviceinstaller" ]
then
    sudo rm /usr/local/bin/ideviceinstaller
fi

exit 0
```

3. コマンドラインで `chmod +x uninstallInfoService.sh` を実行し、シェル ファイルの実行可能にします。
4. コマンドラインからシェル ファイルを実行します。

Google Chrome を使用したテスト

このセクションでは、Google Chrome を使用してテストすることによって、クロス ブラウザー テスト セットを拡張する方法について説明します。

Silk Test Workbench は、Google Chrome 上での操作の記録とテストの再生をサポートします。

- Google Chrome でテストを開始するときに、Google Chrome のインスタンスが実行していない場合、Silk Test Workbench は Google Chrome の新しいインスタンスを開始します。新しいブラウザーは、アドオン無しのキャッシュを空にした状態の一時プロファイルを使用します。
- Google Chrome のインスタンス上でのテストを開始するときに、すでに実行中であれば、Silk Test Workbench はそのインスタンスが最初に開始されたときに使用されたコマンド ライン引数と同じ引

数で Google Chrome を再起動します。この再起動は、Silk Test Workbench オートメーション サポートを有効化するために必要です。

- Google Chrome でテストするとき、Google Chrome インスタンスは、Open Agent のシャットダウン時または、Google Chrome 外のほかのアプリケーションのテストを開始するときに閉じられます。



ヒント: Google Chrome を使用して既存のテスト スクリプトを実行する場合は、基本状態を使用して、URL へ移動するコマンドをテスト スクリプトに追加することを、Micro Focus はお勧めします。

例 1

Google Chrome の実行中のインスタンスが、コマンド「C:/Program Files (x86)/Google/Chrome/Application/chrome.exe www.borland.com」で最初に起動されていた場合、Google Chrome は再起動後に *www.borland.com* を開きます。

例 2

Google Chrome の実行中のインスタンスが、コマンド「C:/Program Files (x86)/Google/Chrome/Application/chrome.exe」で最初に起動されていた場合、Google Chrome は再起動後に *about:blank* を開きます。


Google Chrome を使用したテスト再生の前提条件


コマンドラインパラメータ

Google Chrome を使用してテストの再生やロケーターの記録を行うと、次のコマンドで Google Chrome が起動されます。

```
%LOCALAPPDATA%\%Google%\Chrome%\Application%\chrome.exe
--enable-logging
--log-level=1
--disable-web-security
--disable-hang-monitor
--disable-prompt-on-repost
--dom-automation
--full-memory-crash-report
--no-default-browser-check
--no-first-run
--homepage=about:blank
--disable-web-resources
--disable-preconnect
--enable-logging
--log-level=1
--safebrowsing-disable-auto-update
--test-type=ui
--noerrdialogs
--metrics-recording-only
--allow-file-access-from-files
--disable-tab-closeable-state-watcher
--allow-file-access
--disable-sync
--testing-channel=NamedTestingInterface:st_42
```

ウィザードを使用してアプリケーションに追加する場合は、これらのコマンドラインパラメータは、基本状態に自動的に追加されます。テストを開始したときに、適切なコマンドラインパラメータなしで Google Chrome のインスタンスがすでに実行されている場合、Silk Test Workbench は Google Chrome を終了して、コマンドラインパラメータを使用してブラウザを再起動しようとします。ブラウザを再起動できない場合は、エラーメッセージが表示されます。

 **注:** クロスドメインのドキュメントを記録または再生する場合は、コマンドラインパラメータ `disable-web-security` が必要です。

 **注:** ローカルファイルシステムに保存された Web アプリケーションをテストするには、Google Chrome で `chrome://extensions` に移動し、**Silk Test Chrome Extension** の **ファイルの URL へのアクセスを許可する** チェックボックスをオンにします。


Google Chrome 拡張のテスト

Google Chrome 拡張 (アドオン) を Silk Test Workbench を使ってテストするには、以下の 2 つの手段のいずれかを使用できます。

Google Chrome の起動時に .crx ファイルとして拡張をインストールする

.crx ファイルとしてインストールされた Google Chrome 拡張をテストするには、基本状態に次のコマンドラインを追加します。

```
chrome.exe --load-extension=C:/myExtension/myExtension.crx
```

 **注:** Google Chrome には単一の拡張のみを .crx ファイルとしてインストールできます。Google Chrome に複数の拡張をインストールするには、複数の .crx ファイルをカンマ区切りで指定します。例：

```
chrome.exe --load-extension=C:/myExtension/myExtension.crx,C:/myExtension2/myExtension2.crx
```

ブラウザーのコマンドライン引数指定方法の詳細については、「基本状態の変更」を参照してください。

プロファイルに拡張を追加する

Google Chrome ユーザー データ ディレクトリに拡張を追加し、そのプロファイルをテストに使用します。詳細については、「[ユーザー データ ディレクトリを使用した Google Chrome のテスト](#)」を参照してください。

ユーザー データ ディレクトリを使用した Google Chrome のテスト


ホームページ、使用するツールバーの設定や保存したパスワード、ブックマークなど、Google Chrome で行ったすべての変更は、ユーザー データ ディレクトリと呼ばれる特別なフォルダに格納されます。

Silk Test Workbench を使用して、テスト対象アプリケーションの基本状態にユーザー データ ディレクトリへのパスを指定することによって、Google Chrome ユーザー データ ディレクトリをテストできます。次のコマンドラインには、プロファイルへのパスが含まれています。

```
chrome.exe "--user-data-dir=C:/Users/MyUser/AppData/Local/Google/Chrome/User Data"
```

サンプル Web アプリケーションのプロファイル ディレクトリを設定するには、次のコードを使用できます。

```
' VB code
Dim baseState = New BrowserBaseState(BrowserType.GoogleChrome, "demo.borland.com/InsuranceWebExtJS")
Dim myProfileDir = "--user-data-dir=D:¥¥temp¥¥SilkTest --profile-directory=Profile1"
baseState.CommandLineArguments = myProfileDir
baseState.Execute()
```

 **注:** Google Chrome が Silk Test Workbench によって起動される際には、空のユーザー データ ディレクトリが使用されます。これにより、クリーンな状態でテストが開始されることとなります。

Google Chrome を使用したテストの制限事項

以下のリストに、ローカル Windows マシン上で Google Chrome を使用した場合の、テストの再生とロケーターの記録の既知の制限事項をリストします。

- Silk Test は、Google Chrome を使用した xBrowser ドメインの子テクノロジー ドメインのテストをサポートしていません。たとえば、Apache Flex または Microsoft Silverlight は Google Chrome ではサポートされていません。

- Silk Test Workbench は、HTTP 基本認証ダイアログのテストの記録をサポートしません。
- Silk Test は、Google Chrome のネイティブ サポートは提供しません。内部 Google Chrome 機能をテストすることはできません。たとえば、テストで、Win32 でナビゲーションバーにテキストを追加して現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して Web ページ間を移動できます。Silk Test は、Alert API を使用した警告および類似のダイアログ ボックスの処理をサポートします。
- Silk Test Workbench は Google Chrome を使用して Web アプリケーションをテストする際の列挙子をサポートしていません。
- Silk Test Workbench は、IMoveable クラスの GetFocus メソッドをサポートしません。
- Silk Test は、Google Chrome メニューを使用して Google Chrome の印刷ダイアログ ボックスが開かれたことは認識しません。Google Chrome でダイアログ ボックスを開く動作を追加してテストするには、TypeKeys メソッドを使用して **Ctrl+Shift+P** を送信する必要があります。Internet Explorer はこのショートカットを認識しません。したがって、最初に Internet Explorer でテストを記録してから、手動で **Ctrl+Shift+P** を押す操作をテストに追加する必要があります。
- Google Chrome の複数ウィンドウの同時テストは、最初の Google Chrome ウィンドウから AUT 自身が追加のウィンドウを開いた場合にのみサポートされます。追加の Google Chrome ウィンドウを手動で開いた場合、Silk Test Workbench はこれらの Google Chrome ウィンドウ上の要素を解決できません。たとえば、記録中に AUT のリンクやボタンをクリックして開かれた Google Chrome ウィンドウ上の要素は Silk Test Workbench は解決できますが、記録中に **CTRL+N** を押して開かれた Google Chrome ウィンドウ上の要素は Silk Test Workbench は解決できません。
- ビジュアル テストの再生速度が遅い場合は、**ツール > オプション > 再生 > 結果 > ビジュアル テスト** のオプション **コントロールのキャプチャ** を **いいえ** に設定します。そうしないと、リッチな Web ページでは再生速度が著しく低下し、通信がタイムアウトすることもあります。
- Internet Explorer を使用してテストを再生する場合、executeJavaScript をテストするために次のコードを使用できます。

```
// Java code
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("function foo() { alert('Silk Test'); }");
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("foo();");
```

Google Chrome 上でテストを再生する場合、スクリプトはクロージャで実行され、グローバル コンテキスト (window) では実行されません。すべては関数内で実行されます。上記のサンプル コードの最初の ExecuteJavaScript 呼び出しは、Google Chrome では機能しません。これは、関数 foo が ExecuteJavaScript 呼び出しが存続する間だけ有効であるためです。

Google Chrome 上で同じテストを再生する場合は、次のような関数式を使用できます。

```
// Java code
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("window.foo = function() { alert('Silk Test'); }");
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("window.foo();");
```

前のサンプル コードは Silk4J では機能します。他の Silk Test クライアントに対するコードも同様です。詳細については、Silk Test クライアントのヘルプにある ExecuteJavaScript メソッドのドキュメントを参照してください。

- Google Chrome のユーザー データ ディレクトリをグループ ポリシーを使って設定していると、Google Chrome 上での並列テストは機能しません。この問題を回避するには、レジストリー キー HKEY_LOCAL_MACHINE¥Software¥Policies¥Google¥Chrome¥UserDataDir または HKEY_CURRENT_USER¥Software¥Policies¥Google¥Chrome¥UserDataDir を削除するように、管理者に依頼してください。
- Google Chrome 73 以降のバージョンでは、小さいサイズのブラウザ ウィンドウをテスト モードで開くことができなくなりました。指定可能な最小サイズは、オペレーティング システムに依存します。すべてのオペレーティング システム上で機能する、最小サイズは、550x440 ピクセルです。

macOS 上の Google Chrome を使用したテストの制限事項

以下のリストに、macOS 上で Google Chrome を使用した場合の、テストの再生とロケータの記録の既知の制限事項をリストします。

- Silk Test Workbench は、TypeKeys メソッドで **CMD** キーをサポートしません。
- Silk Test は、Google Chrome を使用した xBrowser ドメインの子テクノロジー ドメインのテストをサポートしていません。たとえば、Apache Flex または Microsoft Silverlight は Google Chrome ではサポートされていません。
- Silk Test Workbench は、HTTP 基本認証ダイアログのテストの記録をサポートしません。
- Silk Test は、Google Chrome のネイティブ サポートは提供しません。内部 Google Chrome 機能をテストすることはできません。たとえば、テストで、Win32 でナビゲーションバーにテキストを追加して現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して Web ページ間を移動できます。Silk Test は、Alert API を使用した警告および類似のダイアログ ボックスの処理をサポートします。
- Silk Test Workbench は、IMoveable クラスの GetFocus メソッドをサポートしません。
- 複数の Google Chrome ウィンドウ上での同時テストは、macOS 上ではサポートされません。
- 既に開いている Google Chrome ウィンドウへのアタッチは、macOS 上ではサポートされません。
- Internet Explorer を使用してテストを再生する場合、executeJavaScript をテストするために次のコードを使用できます。

```
// Java code
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("function foo() { alert('Silk Test'); }");
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("foo();");
```

Google Chrome 上でテストを再生する場合、スクリプトはクロージャで実行され、グローバル コンテキスト (window) では実行されません。すべては関数内で実行されます。上記のサンプルコードの最初の ExecuteJavaScript 呼び出しは、Google Chrome では機能しません。これは、関数 foo が ExecuteJavaScript 呼び出しが存続する間だけ有効であるためです。

Google Chrome 上で同じテストを再生する場合は、次のような関数式を使用できます。

```
// Java code
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("window.foo = function() { alert('Silk Test'); }");
desktop.<BrowserWindow> find("//BrowserWindow")
    .executeJavaScript("window.foo();");
```

前のサンプルコードは Silk4J では機能します。他の Silk Test クライアントに対するコードも同様です。詳細については、Silk Test クライアントのヘルプにある ExecuteJavaScript メソッドのドキュメントを参照してください。


Mozilla Firefox を使用したテスト

このセクションでは、Mozilla Firefox を使用してテストすることによって、クロス ブラウザー テスト セットを拡張する方法について説明します。

Silk Test Workbench は、Mozilla Firefox 上での操作の記録とテストの再生をサポートします。

- Mozilla Firefox でテストを開始するときに、Mozilla Firefox のインスタンスが実行していない場合、Silk Test Workbench は Mozilla Firefox の新しいインスタンスを開始します。新しいブラウザーは、アドオン無しのキャッシュを空にした状態の一時プロファイルを使用します。
- Mozilla Firefox のインスタンス上でのテストを開始するときに、すでに実行中であれば、Silk Test Workbench はそのインスタンスが最初に開始されたときに使用されたコマンド ライン引数と同じ引数で Mozilla Firefox を再起動します。この再起動は、Silk Test Workbench オートメーション サポートを有効化するために必要です。

- この Mozilla Firefox のインスタンスは、Open Agent のシャットダウン時または、Mozilla Firefox 外のほかのアプリケーションのテストを開始するときに閉じられます。

 **ヒント:** Mozilla Firefox を使用して既存のテスト スクリプトを実行する場合は、基本状態を使用して、URL へ移動するコマンドをテスト スクリプトに追加することを、Micro Focus はお勧めします。

Mozilla Firefox で記録する場合、Mozilla Firefox は外部リンクを新しいウィンドウではなく新しいタブで開きます。外部リンクを新しいウィンドウで開く場合は、**詳細設定** オプションの **Firefox のシングル ウィンドウ モードを使用する** オプションを無効にします。

例 1

Mozilla Firefox の実行中のインスタンスが、コマンド「C:/program files/Mozilla/firefox.exe www.borland.com」で最初に起動されていた場合、Mozilla Firefox は再起動後に *www.borland.com* を開きます。

例 2

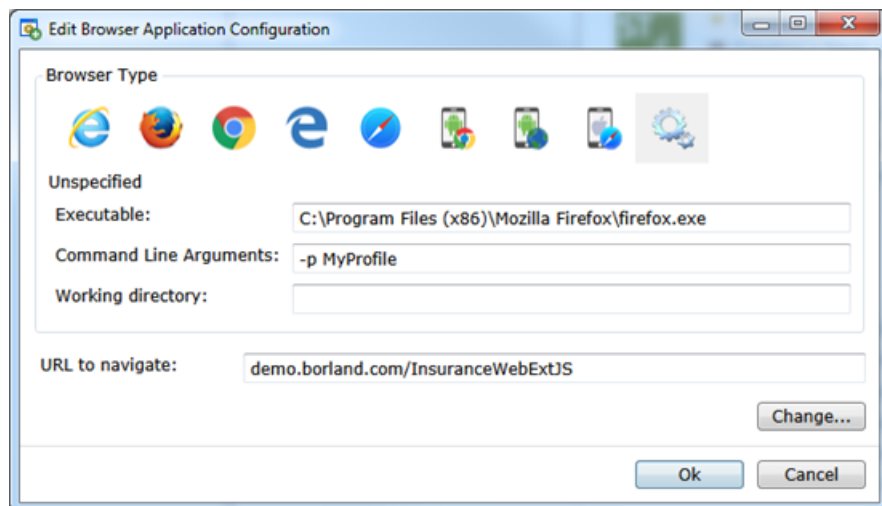
Mozilla Firefox の実行中のインスタンスが、コマンド「C:/program files/Mozilla/firefox.exe」で最初に起動されていた場合、Mozilla Firefox は再起動後に *about:blank* を開きます。

プロファイルを使用した Mozilla Firefox のテスト

ホームページ、使用するツールバーの設定や保存したパスワード、ブックマークなど、Mozilla Firefox で行ったすべての変更は、プロファイルと呼ばれる特別なフォルダに格納されます。

Mozilla Firefox プロファイル进行测试するには、プロファイルの名前、またはプロファイルへのパスのどちらかをカスタム ブラウザー コマンド ライン引数として **アプリケーション構成の編集** ダイアログ ボックスで指定します。

1. Mozilla Firefox プロファイル进行测试するプロジェクトを選択します。
2. メニューから **アプリケーション構成の編集** をクリックします。**アプリケーション構成の編集** ダイアログ ボックスが表示されます。
3. **編集** をクリックして、既存の **ブラウザー : Firefox** アプリケーション構成を編集します。
4. ブラウザーの種類から **カスタム** を選択します。
5. **実行可能ファイル** フィールドに、Mozilla Firefox 実行可能ファイルへのパスを入力します。たとえば、C:\Program Files (x86)\Mozilla Firefox\firefox.exe など。
6. **コマンド ライン引数** フィールドに、Firefox プロファイル名または Firefox プロファイルへのパスを入力します。たとえば、-p myProfile または -profile C:/Temp など。




プロファイルの名前の使用 次のコマンド ライン引数では、プロファイル名を指定しています。


```
-p myProfile
```

プロファイル マネージャ で名前をつけてプロファイルを設定できます。**プロファイル マネージャ** を起動するには、コマンド ウィンドウで `firefox.exe -P` を入力します。

プロファイルへのパスの使用 次のコマンド ライン引数では、プロファイルへのパスを指定しています。

```
-profile C:/<path to profile folder>
```

 **注:** Silk Test Workbench が Mozilla Firefox を起動する場合、空のプロファイルが使用されます。これにより、クリーンな状態でテストが開始されることとなります。ユーザーが手動で Mozilla Firefox を起動した場合は、デフォルトのプロファイルが使用されます。

 **注:** プロファイルはテストマシンにデプロイする必要があります。メモリ消費量も大きいので、プロファイルのテスト時に問題が発生する可能性があります。いくつかのブラウザーの設定を変更するだけであれば、プロファイルの代わりにキーバディリティを使用できます。詳細については、「[WebDriver ベースのブラウザーのキーバディリティの設定](#)」を参照してください。

Mozilla Firefox 拡張機能のテスト

Silk Test Workbench を使って Mozilla Firefox 拡張機能（アドオン）をテストするには、Mozilla Firefox プロファイルに拡張機能を追加し、そのプロファイルをテストに使用します。詳細については、「[プロファイルを使用した Mozilla Firefox のテスト](#)」を参照してください。

Mozilla Firefox を使用したテストの制限事項

Silk Test Workbench を使用して Mozilla Firefox 上の Web アプリケーションをテストする際の既知の制限事項を以下に示します。

- Mozilla Firefox の複数ウィンドウの同時テストは、最初の Mozilla Firefox ウィンドウから AUT 自身が追加のウィンドウを開いた場合にのみサポートされます。追加の Mozilla Firefox ウィンドウを手動で開いた場合、Silk Test Workbench はこれらの Mozilla Firefox ウィンドウ上の要素を解決できません。たとえば、記録中に AUT のリンクやボタンをクリックして開かれた Mozilla Firefox ウィンドウ上の要素は Silk Test Workbench は解決できますが、記録中に **CTRL+N** を押して開かれた Mozilla Firefox ウィンドウ上の要素は Silk Test Workbench は解決できません。
- Silk Test Workbench は `window.showmodaldialog` コマンドを使って表示されるウィンドウであるモーダル ブラウザー ウィンドウのテストをサポートしません。これらのモーダル ブラウザー ウィンドウは、公式に廃止されており、Google Chrome 37 以降では無効化されています。さらに、Mozilla Firefox の今後のバージョンではサポートされる計画はありません。低レベルな操作を使ってこの問題を回避できます。たとえば、オブジェクトの座標を使用したネイティブなクリックやテキスト フィールドを `typeKeys` を使って入力する方法が利用できます。
- Silk Test Workbench は、Mozilla Firefox を使用した Silverlight のテストをサポートしません。
- Silk Test Workbench は Mozilla Firefox の **バージョン情報** ダイアログなどの一部のブラウザー ダイアログのテストをサポートしません。
- Silk Test Workbench は、Mozilla Firefox を使用した `about:*` ページのテストをサポートしません。
- Silk Test Workbench は、Mozilla Firefox の **印刷** ボタンのクリックの記録をサポートしません。再生時にこのボタンをクリックする場合は、座標によるデスクトップのクリックをテスト スクリプトに手動で追加することにより行えます。例：

```
'VB .NET code
_desktop.Click(printButton.GetRect(true).Center)
```
- Silk Test は、Mozilla Firefox のネイティブ サポートは提供しません。内部 Mozilla Firefox 機能をテストすることはできません。たとえば、テストで、Win32 でナビゲーション バーにテキストを追加して現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して Web ページ間を移動できます。Silk Test は、Alert API を使用した警告および類似のダイアログ ボックスの処理をサポートします。

- Silk Test Workbench は Mozilla Firefox の JavaScript 警告ボックスのロケータの記録をサポートしません。さらに、Mozilla Firefox を使用して JavaScript 警告ボックスを処理するために、次のメソッドを使用することはできません。
 - AcceptAlert
 - DismissAlert
 - GetAlertText
 - IsAlertPresent
- Silk Test Workbench は、Mozilla Firefox では Java アプレットをサポートしません。
- Silk Test Workbench は、Mozilla Firefox では XPath 式のプロパティをサポートしません。XPath 式では、属性のみがサポートされます。
- Silk Test Workbench は、IMoveable クラスの GetFocus メソッドをサポートしません。
- Silk Test は、Mozilla Firefox を使用した xBrowser ドメインの子テクノロジー ドメインのテストをサポートしていません。たとえば、Apache Flex または Microsoft Silverlight は Mozilla Firefox ではサポートされていません。
- Mozilla Firefox では、次のメソッドはサポートされません。
 - PressKeys
 - ReleaseKeys
- Mozilla Firefox では、次のネイティブ再生はサポートされません。
 - ダブルクリック。
 - マウス ボタンの右と中央のクリック。
- Mozilla Firefox では、DomClick メソッドは、警告を開くコントロールではサポートされません。

macOS 上の Mozilla Firefox を使用したテストの制限事項

Mozilla Firefox を使用して macOS 上の Web アプリケーションをテストする際の既知の制限事項を以下に示します。

- Silk Test Workbench は、macOS 上の Mozilla Firefox 54 以降を使用してテストされました。
- Mozilla Firefox の複数ウィンドウの同時テストは、最初の Mozilla Firefox ウィンドウから AUT 自身が追加のウィンドウを開いた場合にのみサポートされます。追加の Mozilla Firefox ウィンドウを手動で開いた場合、Silk Test Workbench はこれらの Mozilla Firefox ウィンドウ上の要素を解決できません。たとえば、記録中に AUT のリンクやボタンをクリックして開かれた Mozilla Firefox ウィンドウ上の要素は Silk Test Workbench は解決できますが、記録中に **CTRL+N** を押して開かれた Mozilla Firefox ウィンドウ上の要素は Silk Test Workbench は解決できません。
- Silk Test Workbench は window.showmodalDialog コマンドを使って表示されるウィンドウであるモーダルブラウザー ウィンドウのテストをサポートしません。これらのモーダルブラウザー ウィンドウは、公式に廃止されており、Google Chrome 37 以降では無効化されています。さらに、Mozilla Firefox の今後のバージョンではサポートされる計画はありません。低レベルな操作を使ってこの問題を回避できます。たとえば、オブジェクトの座標を使用したネイティブなクリックやテキスト フィールドを typeKeys を使って入力する方法が利用できます。
- Silk Test Workbench は、Mozilla Firefox を使用した Silverlight のテストをサポートしません。
- Silk Test Workbench は Mozilla Firefox の **バージョン情報** ダイアログなどの一部のブラウザー ダイアログのテストをサポートしません。
- Silk Test Workbench は、Mozilla Firefox を使用した about:* ページのテストをサポートしません。
- Silk Test Workbench は、Mozilla Firefox の **印刷** ボタンのクリックの記録をサポートしません。再生時にこのボタンをクリックする場合は、座標によるデスクトップのクリックをテスト スクリプトに手動で追加することにより行えます。例：


```
'VB .NET code
_desktop.Click(printButton.GetRect(true).Center)
```
- Silk Test は、Mozilla Firefox のネイティブ サポートは提供しません。内部 Mozilla Firefox 機能をテストすることはできません。たとえば、テストで、Win32 でナビゲーション バーにテキストを追加して現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して

Web ページ間を移動できます。Silk Test は、Alert API を使用した警告および類似のダイアログ ボックスの処理をサポートします。

- Silk Test Workbench は Mozilla Firefox の JavaScript 警告ボックスのロケータの記録をサポートしません。さらに、Mozilla Firefox を使用して JavaScript 警告ボックスを処理するために、次のメソッドを使用することはできません。
 - AcceptAlert
 - DismissAlert
 - GetAlertText
 - IsAlertPresent
- Silk Test Workbench は、macOS 上の Mozilla Firefox では Java アプレットをサポートしません。
- Silk Test Workbench は、Mozilla Firefox では XPath 式のプロパティをサポートしません。XPath 式では、属性のみがサポートされます。
- Silk Test Workbench は、IMoveable クラスの GetFocus メソッドをサポートしません。
- Silk Test は、Mozilla Firefox を使用した xBrowser ドメインの子テクノロジー ドメインのテストをサポートしていません。たとえば、Apache Flex または Microsoft Silverlight は Mozilla Firefox ではサポートされていません。
- 次のメソッドはサポートされません。
 - PressKeys
 - ReleaseKeys
- 次のネイティブ再生はサポートされません。
 - ダブルクリック。
 - マウス ボタンの右と中央のクリック。
- DomClick メソッドは、警告を開くコントロールではサポートされません。
- Mozilla Firefox 55 を使用する場合、ファイルのアップロードは機能しません。詳細については、『[File upload no longer works with geckodriver 0.18.0 and Firefox 55](#)』を参照してください。

Microsoft Edge を使用したテスト

このセクションでは、Microsoft Edge を使用してテストすることによって、クロス ブラウザー テスト セットを拡張する方法について説明します。

Microsoft Edge を使用したテストの制限事項

以下に、Microsoft Edge を使用してテストする際の既知の制限事項を一覧します。

- 次のクラス、インターフェイス、メソッド、プロパティは、Microsoft Edge 上の Web アプリケーションのテストでは現時点ではサポートされません：
 - BrowserApplication クラス。
 - ClearCache メソッド
 - CloseOtherTabs メソッド
 - CloseTab メソッド
 - ExistsTab メソッド
 - GetHorizontalScrollbar メソッド
 - GetNextCloseWindow メソッド
 - GetSelectedTab メソッド
 - GetSelectedTabIndex メソッド
 - GetSelectedTabName メソッド
 - GetTabCount メソッド
 - GetVerticalScrollbar メソッド
 - IsActive メソッド

- Minimize メソッド
- OpenContextMenu メソッド
- OpenTab メソッド
- Restore メソッド
- SelectTab メソッド
- SetActive メソッド
- WindowState メソッド
- BrowserWindow クラスの次のメソッドは、Microsoft Edge のビルド 38.14393 (Microsoft Windows 10 Anniversary Update のバージョンの Microsoft Edge) より前のバージョンではサポートされません。
 - PressKeys メソッド
 - ReleaseKeys メソッド
- Silk Test Workbench は、Microsoft Edge に対する操作を記録するとき、ブラウザを自動的に最前面に表示しません。
- Microsoft Edge を使用してテストすると、BrowserApplication の矩形領域は絶対値ではありません。
- Silk Test Workbench は、Apache Flex のテストをサポートしません。
- Silk Test Workbench は、Microsoft Edge のネイティブ サポートは提供しません。内部 Microsoft Edge 機能をテストすることはできません。たとえば、テストで、ナビゲーションバーにテキストを追加して、現在表示されている Web ページを変更することはできません。回避策として、API コールを使用して Web ページ間を移動できます。
- Silk Test Workbench は、Microsoft Edge の警告ダイアログなどのダイアログ ボックスをサポートしません。
- イメージ クリックは、Microsoft Edge Threshold 2 (ビルド 25.10586) 以降に対してのみサポートされます。Microsoft Edge の以前のバージョン上で Web アプリケーションをテストする場合は、イメージ検証のみ使用できます。
- Silk Test Workbench は、Microsoft Edge のタブをサポートしません。タブはウィンドウとして解決されます。
- ビジュアル テストの再生速度が遅い場合は、**ツール > オプション > 再生 > 結果 > ビジュアル テスト** のオプション **コントロールのキャプチャ** を **いいえ** に設定します。そうしないと、リッチな Web ページでは再生速度が著しく低下し、通信がタイムアウトすることもあります。
- Microsoft Edge 上の Web アプリケーションをテストするとき、Silk Test Workbench は http-equiv 属性などのメタタグを検索できません。たとえば、Silk Test Workbench は次のメタタグを検索できません：


```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```
- Microsoft Edge を使用する場合、Silk Test Workbench は DOM 要素の currentStyle 属性の直接読み取りをサポートしません。DomElement クラスの GetCssStyle メソッドを使用して、指定したスタイル名の算出 CSS スタイルを取得できます。
- Microsoft Edge 上の Web アプリケーションとのやり取りを開始するとき、Silk Test Workbench は Microsoft Edge のすべての開いているインスタンスを閉じ、新しくブラウザを開始します。新しいブラウザは、アドオン無しのキャッシュを空にした状態の一時プロファイルを使用します。この Microsoft Edge のインスタンスは、Open Agent のシャットダウン時または、Microsoft Edge 外のほかのアプリケーションのテストを開始するときに閉じられます。
- Microsoft Edge を使用する場合、Silk Test Workbench は操作やロケータの記録中に textContents 属性を認識しません。しかし、オブジェクト マップで textContents 属性を使用して、Microsoft Edge 上でテストを再生するときに使用できます。
- Silk Test Workbench は、Microsoft Edge では XPath 式のプロパティをサポートしません。XPath 式では、属性のみがサポートされます。
- Silk Test Workbench は、IMoveable クラスの GetFocus メソッドをサポートしません。
- Silk Test Workbench は、Content-Security-Policy HTTP ヘッダーを含んだ Web アプリケーションのテストをサポートしません。

レスポンシブ Web デザインのテスト

レスポンシブ Web デザインに基づいて構築されたデスクトップ Web アプリケーションは、アプリケーションを表示する画面や Web ブラウザーのサイズに応じて、その外観が変化します。このようなテストでは、再生するウィンドウのサイズを適切に選択できないと、その安定性に多大な影響を与える可能性があります。

Silk Test Workbench では、次のタイミングでブラウザー ウィンドウの正確なサイズを指定することができます。

- Web アプリケーションに対して新しいプロジェクトを作成する時点
- 新しいビジュアル テストや .NET スクリプトを作成する時点
- 新しいテストを記録する時点
- 既存のテストに操作を記録する時点
- Web アプリケーションに対するテストを再生する時点

次の設定を使用して、ブラウザー ウィンドウのサイズを指定することができます。

- **ブラウザー サイズ** リストには、あらかじめ定義されたブラウザー ウィンドウのサイズとカスタマイズしたサイズの両方が表示され、ここからテストで使用するサイズを選択できます。
- **向き** リストでは、ブラウザー ウィンドウの向きが横向きか縦向きかを選択できます。
- **ブラウザー サイズの編集** をクリックすると、**ブラウザー サイズ** リストにカスタム ブラウザー サイズを追加したり、リストに表示するブラウザー サイズを選択することができます。
 - 新しいカスタム ブラウザー サイズをリストに追加するには、**ブラウザー サイズの追加** をクリックします。
 - リストにサイズを表示しないようにするには、対応するチェック ボックスをオフにします。
 - 特定の Web アプリケーションのビジュアル ブレークポイントを **ブラウザー サイズ** リストに追加するには、**ビジュアルブレークポイントの検出** をクリックします。

コマンド ラインや Silk Central から Web アプリケーションに対するテストを再生する場合は、`silktest.browserViewportSize` 環境変数を設定してブラウザー ビューポートのサイズを指定できます。**ブラウザー サイズ** リストで定義したブラウザーの名前や特定のサイズを指定することができます。

Imports SilkTest.ntf.Mobile をテスト スクリプトで定義すると、テストスクリプトで直接ブラウザーのサイズを変更できます：

```
' VB .NET code
With .BrowserWindow("BrowserWindow")
    .SetViewportName("Google Pixel 2")
    .SetViewportSize(800, 300)
End With
```

例：自動再生するブラウザー サイズに名前を使用して指定する場合

ブラウザー サイズ リストの SVGA エントリを使用して、ブラウザー サイズを SVGA (800, 600) に設定するサンプルコードを以下に示します。

```
SET silktest.browserViewportSize=name=SVGA;orientation=landscape
STW.exe -script MyScript
```

次のコマンドを使用して、特定のブラウザーとビューポート サイズを指定することもできます。

```
STW -dsn silktest -browser InternetExplorer -script MyScript -
viewportname SVGA
```

例：自動再生するブラウザー サイズに幅と高さを使用して指定する場合

width と height パラメータを使用して、ブラウザー サイズを SVGA (800, 600) に設定するサンプルコードを以下に示します。

```
SET  
silktest.browserViewportSize=width=800;height=600;orientation=landscape  
STW.exe -script MyScript
```

次のコマンドを使用して、特定のブラウザーとビューポート サイズを指定することもできます。

```
STW -dsn silktest -browser InternetExplorer -script MyScript -  
viewportwidth 800 -vieportheight 600
```

ビジュアルブレイクポイントの検出

レスポンシブ Web アプリケーションのビジュアルブレイクポイントを検出する前に、Mozilla Firefox 56 以降または Google Chrome 60 以降を Silk Test Workbench を実行しているマシン上にインストールしてください。

レスポンシブ Web デザイン技術を使用して実装する多くの Web アプリケーションは、表示されるブラウザーやデバイスのサイズに応じて、そのレイアウトを変更します。レイアウトの変更が発生する特定の解像度を、ビジュアルブレイクポイントと言います。

Silk Test Workbench では、ビジュアルブレイクポイントを検出し、それを使用して特定のサイズに記録ウィンドウのサイズを変更することで、このようなアプリケーションのテストをサポートします。

Silk Test Workbench では、次のタイミングでブラウザー ウィンドウの正確なサイズを指定することができます。

- Web アプリケーションに対して新しいプロジェクトを作成する時点
- 新しいビジュアルテストや .NET スクリプトを作成する時点
- 新しいテストを記録する時点
- 既存のテストに操作を記録する時点
- Web アプリケーションに対するテストを再生する時点

Web アプリケーションのビジュアルブレイクポイントを検出し、対応する解像度を **ブラウザー サイズ** リストに表示させるには、次の手順を実行します。

1. **ブラウザー サイズの編集** をクリックします。 **ブラウザー サイズの編集** ダイアログ ボックスが表示されます。
2. **ビジュアルブレイクポイントの検出** をクリックします。 Web アプリケーションの URL がアプリケーション構成や基本状態で指定されていない場合は、 **ビジュアルブレイクポイント検出 URL** ダイアログが表示されます。
3. Web アプリケーションの URL がアプリケーション構成や基本状態で指定されていない場合は、 **ビジュアルブレイクポイント検出 URL** ダイアログが表示されます。 URL をテキスト フィールドに入力し、 **OK** をクリックします。 Silk Test Workbench は Web アプリケーションのすべてのビジュアルブレイクポイントを検出し、 **ブラウザー サイズ** リストに追加します。
4. **OK** をクリックして **ブラウザー サイズの編集** ダイアログを閉じます。

以上により、テストするブラウザー ウィンドウまたはモバイル デバイスのサイズに合わせて、ビジュアルブレイクポイントを選択できるようになります。

iframe パフォーマンスの向上

Internet Explorer 以外のブラウザ上で、多くの iframe を使用した Web アプリケーションに対するテストを再生すると、速度が遅くなる場合があります。このトピックでは、次のブラウザ使用時における、パフォーマンスを最適化する方法について、いくつかの方法を提案します。

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Apple Safari
- サポートするモバイル ブラウザー（モバイル デバイスまたはエミュレータ上）

このようなパフォーマンスの問題に直面した場合、以下に示すパフォーマンス最適化手法をテスト スクリプトに対して試すことができます。

たとえば、以下のような iframe 構造を持つ Web アプリケーションをテストしていることを想定します。


```
BrowserWindow
  iframe name=1
    iframe name=2
    iframe name=3
    iframe name=4
    iframe name=5
  iframe name=6
    iframe name=7
    iframe name=8
    iframe name=9
```

この場合、以下のようなパフォーマンス最適化手法を試すことができます。

iframe サポートの包含リストを使って、テスト対象にする iframe のみを有効化する

たとえば、上記の iframe 構造において、name が 8 の iframe 内の要素だけを操作対象とする場合を考えます。このような要素に対するロケータは次のようになります。

```
//BrowserApplication//BrowserWindow//IFRAME[@name='6']//IFRAME[@name='8']//input[@name='username']
```

 **注:** Internet Explorer 以外のすべてのサポートするブラウザに対するロケータには、iframe が含まれます。

name が 8 の iframe の要素を操作対象とするには、すべての親 iframe も **iframe サポートの包含リスト** に指定する必要があります。この例の場合は、「name:6, name:8」を **iframe サポートの包含リスト** フィールドに指定します。

iframe サポートの除外リストを使って、テスト対象にしない iframe を無効化する

iframe サポートの除外リスト を使って、広告を表示する iframe などを無効化して、テスト対象からはずすことができます。

たとえば、上記の iframe 構造において、name が 1 の iframe とその子 iframe の要素をテストする必要が無ければ、「name:1」を **iframe サポートの除外リスト** フィールドに指定します。

iframe サポートを完全に無効化する

iframe サポートの有効化 オプションに **いいえ** を設定して、iframe サポートを完全に無効化することができます。

たとえば、アプリケーションのすべての iframe が広告を表示するために使用されている場合、これらの iframe をテストする必要が無ければ、このオプションを使用できます。

iframe の検索数を低減するロケータ一の構成

iframe の検索数を減らすようにロケータを構成すると、検索操作の再生パフォーマンスを向上させることができます。このパフォーマンス最適化手法は、**iframe サポートの除外リスト** や **iframe サポートの包含リスト** と組み合わせて使用すると、より効果的です。

たとえば、上記の iframe 構造において、name が 8 の iframe 内の要素だけを操作対象とする場合を考えます。XPath ロケータやオブジェクト マップ エントリを、iframe の検索数を減らすように構成することができます。

次のサンプル スクリプトでは、name が 8 の iframe を複数回検索します。

```
' VB code
With _desktop.BrowserApplication("//BrowserApplication")
  With .BrowserWindow("//BrowserWindow")
    .DomElement("//IFRAME[@name='6']//IFRAME[@name='8']//
input[@name='username']").TypeKeys("my user name")
    .DomElement("//IFRAME[@name='6']//IFRAME[@name='8']//
input[@name='password']").TypeKeys("top secret")
    .DomElement("//IFRAME[@name='6']//IFRAME[@name='8']//
input[@name='loginButton']").Click()
  End With
End With
```

次のようにロケータを構成することにより、name が 8 の iframe を一度だけ検索するように変更できます。

```
' VB code
With _desktop.BrowserApplication("//BrowserApplication")
  With .BrowserWindow("//BrowserWindow")
    With .DomElement("//IFRAME[@name='6']//IFRAME[@name='8']")
      .DomElement("//input[@name='username']").TypeKeys("my user
name")
      .DomElement("//input[@name='password']").TypeKeys("top
secret")
      .DomElement("//input[@name='loginButton']").Click()
    End With
  End With
End With
```

追加のブラウザのバージョンでのテスト

このトピックでは、Silk Test Workbench のバージョンに自動的に含まれない WebDriver ベースのブラウザの追加のバージョンでテストを行う方法について説明します。



注: Silk Test Workbench は、ブラウザの最新のバージョンを自動的にサポートします。ただし、ブラウザのベンダーが Silk Test Workbench のアップデートを必要とする変更を行った場合、このトピックに記述した手順を行っても、新しいブラウザのバージョンでテストを行うことができない場合もあります。

このトピックで説明されている機能は、次のブラウザに対してサポートされます。

- Google Chrome
- Microsoft Edge
- Mozilla Firefox

1. テストするブラウザのバージョンに対して適切なドライバーをダウンロードします。

- Google Chrome の場合、追加の ChromeDriver のバージョンは [Downloads - ChromeDriver](#) からダウンロードできます。

- Mozilla Firefox の場合、追加の geckodriver のバージョンは [Releases - mozilla/geckodriver](#) からダウンロードできます。
 - Microsoft Edge の場合、追加の Microsoft WebDriver のバージョンは [WebDriver - Microsoft Edge](#) からダウンロードできます。
2. Silk Test Workbench インストール フォルダの、¥ng¥WebDrivers¥ フォルダに移動します。
 3. 使用する新しいブラウザのバージョンのオペレーティング システムに対応するフォルダを開きます。
 - Microsoft Windows の場合、Windows フォルダを開きます。
 - macOS の場合、osx64 フォルダを開きます。
 4. ブラウザーに対して適切なフォルダを開きます。
 - Google Chrome の場合、Chrome を開きます。
 - Mozilla Firefox の場合、Gecko を開きます。
 - Microsoft Edge の場合、Edge を開きます。
 5. 新しいドライバーのバージョン用に新しいフォルダを作成します。
たとえば、ドライバーが ChromeDriver 2.26 であれば、2.26 という新しいフォルダを作成します。
 6. ダウンロードしたドライバーを新しいフォルダで展開します。
 7. Silk Test Workbench インストール フォルダで、¥ng¥WebDrivers¥Common¥Config¥ フォルダに移動します。
 8. ブラウザーに対して適切なフォルダを開きます。
 - Google Chrome の場合、Chrome を開きます。
 - Mozilla Firefox の場合、Gecko を開きます。
 - Microsoft Edge の場合、Edge を開きます。
 9. プロパティ ファイルをテキスト エディターで開きます。
たとえば、Google Chrome の場合、Chrome.properties を開きます。
 - 10 新しいブラウザのバージョンと新しいドライバーのバージョンを次のように追加します。

```
<browser version>=<driver version>
```

たとえば、Google Chrome 53 でテストする場合、ChromeDriver 2.26 が必要であるため、Chrome.properties ファイルに次の行を追加する必要があります。

```
53=2.26
```
 - 11 プロパティ ファイルを保存します。

クロスブラウザ テスト:よくある質問

このセクションでは、さまざまなブラウザ上で Web アプリケーションをテストするときが発生することがある質問を示します。

再生中にダイアログが認識されない

スクリプトを記録するときに、Silk Test Workbench はいくつかのウィンドウを Dialog として認識しません。スクリプトをクロス ブラウザ スクリプトとして使用する場合は、ブラウザによっては Dialog が認識されないため、Dialog を Window に置き換える必要があります。

たとえば、スクリプトに以下の行があるとしてします。

```
/BrowserApplication//Dialog//PushButton[@caption='OK']
```

クロス ブラウザ テストを可能にするには、次のように行を書き換えます。

```
/BrowserApplication//Window//PushButton[@caption='OK']
```

DomClick(x, y) が Click(x, y) のように動作しない

アプリケーションで onclick イベントを使用しており、座標を必要とする場合、DomClick メソッドは動作しません。代わりに、Click を使用します。

FileInputField.DomClick() でダイアログが開かない

代わりに、Click を使用します。

テストの開始時にブラウザ ウィンドウを最大化する方法

テストの開始時など、テスト スクリプトからブラウザを最大化するために、BrowserApplication クラスの Maximize メソッドを使用できます。

ブラウザを最大化するには、テスト スクリプトに次のコードを追加します。

```
'VB code
With _desktop.BrowserApplication("demo_borland_com")

    .Maximize()

End With
```

ブラウザをスクロールさせる方法

VB .NET の場合、Silk Test Workbench では、次のようにして再生中にブラウザの表示領域にコントロールをスクロールさせることができます。

ExecuteJavaScript メソッド (DomElement) ScrollIntoView メソッドを使用して、特定の DOM 要素をブラウザ ウィンドウの表示領域にスクロールできます。

ExecuteJavaScript メソッド (BrowserWindow) ExecuteJavaScript メソッドを使用して、範囲を指定してページ全体を上下にスクロールさせることができます。

使用例

次のコマンドは、1 ページ下にスクロールさせます。

```
'VB .NET code
browserWindow.ExecuteJavaScript("window.scrollTo(0, window.innerHeight)")
```

次のコマンドは、100 ピクセル下にスクロールさせます。

```
'VB .NET code
browserWindow.ExecuteJavaScript("window.scrollTo(0, 100)")
```

次のコマンドは、100 ピクセル上にスクロールさせます。

```
'VB .NET code
browserWindow.ExecuteJavaScript("window.scrollTo(0, -100)")
```

現在使用しているブラウザの確認方法

BrowserApplication クラスには、ブラウザの種類を返すプロパティ "browsertype" があります。このプロパティをロケータに追加することで、どのブラウザに一致させるかを定義できます。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリースノート](#)』を参照してください。

使用例

ブラウザの種類を取得するには、次のコードをロケーターに入力します。
`browserApplication.GetProperty("browserType")`

また、BrowserWindow には、現在のウィンドウのユーザー エージェント文字列を返すメソッド GetUserAgent があります。

要素のテキストに使用されるフォント タイプの確認方法

属性名を「:」で区切ると、DOM 要素の currentStyle 属性のすべての属性にアクセスできます。

Internet Explorer 8 以前 wDomElement.GetProperty("currentStyle:fontName")

Internet Explorer 9 またはそれ以降および Mozilla Firefox などの他のすべてのブラウザ wDomElement.GetProperty("currentStyle:font-name")

innerText をカスタム クラス属性として構成したが、ロケーターで使用されない

ロケーター文字列に使用する属性には最大長があります。InnerText は長くなりすぎる傾向があり、ロケーターで使用できない場合があります。可能な場合は、textContent を代わりに使用してください。

xBrowser API で公開されていない機能が必要な場合の対処方法

ExecuteJavaScript() を使用して、JavaScript コードを Web アプリケーションから直接実行できます。この方法は、ほとんどすべての問題の回避策となります。

Link.Select で、Internet Explorer で新しく開いたウィンドウにフォーカスが設定されない

この制限は、ブラウザの構成設定を変更することで修正できます。新しく開いたウィンドウが常にアクティブ化されるようにオプションを設定します。

アプリケーションのログ出力に正しくないタイムスタンプが含まれる

この方法によって、同期に関して予期しない結果が発生する場合があります。この問題を回避するには、HTML 同期モードを指定します。

新しいページに移動したあと、スクリプトがハングする

この問題は、AJAX アプリケーションによりブラウザがビジー（サーバー プッシュ/ActiveX コンポーネントの接続が開いている）のままになっている場合に、発生することがあります。HTML 同期モードを設定してください。他のトラブルシューティングのヒントについては、「xBrowser のページ同期」のトピックを参照してください。

正しくないロケーターが記録されている

マウスを要素上に移動したときに、要素の属性が変更することがあります。Silk Test Workbench によってこのシナリオの追跡が試行されますが、失敗することがあります。影響を受ける属性を特定し、それが Silk Test Workbench で無視されるように構成してください。

Internet Explorer で要素を囲む四角形の位置が正しくない

- 拡大率が 100% に設定されていることを確認します。このようにしないと、四角形が正しく配置されません。

- ブラウザ ウィンドウの上に通知バーが表示されていないことを確認します。 Silk Test Workbench では、通知バーを処理できません。

マウス移動設定がオンになっているにもかかわらず、すべての操作が記録されない理由

多くの無用な MoveMouse 操作がスクリプトに影響を及ぼさないように、Silk Test Workbench では以下の操作が行われます。

- マウスが一定時間静止している場合にのみ、MoveMouse 操作が記録されます。
- マウスを要素上に移動したあとで操作が行われていることが確認された場合にのみ、MoveMouse 操作が記録されます。場合によっては、スクリプトに手動操作を追加することが必要となることがあります。
- Silk Test Workbench は、Web アプリケーション、Win32 アプリケーション、および Windows Forms アプリケーションに対してのみ、マウス移動の記録をサポートします。Silk Test Workbench は、Apache Flex や Swing など、xBrowser テクノロジ ドメインの子テクノロジ ドメインのマウス移動を記録することはできません。

textContent、innerText、および innerHtml の違い


- textContents は、書式設定のみを目的とする要素およびその子要素に含まれるすべてのテキストです。
- innerText は、要素およびその子要素に含まれるすべてのテキストを返します。
- innerHtml は、要素に含まれるすべてのテキスト (html タグも含む) を返します。

以下の html コードについて検討します。

```
<div id="mylinks">
  This is my <b>link collection</b>:
  <ul>
    <li><a href="www.borland.com">Bye bye <b>Borland</b> </a></li>
    <li><a href="www.microfocus.com">Welcome to <b>Micro Focus</b></a></li>
  </ul>
</div>
```

以下の表に、返されるプロパティの詳細を示します。

コード	返される値
browser.DomElement("//div[@id='mylinks']").GetProperty("textContent")	This is my link collection:
browser.DomElement("//div[@id='mylinks']").GetProperty("innerText")	This is my link collection:Bye bye Borland Welcome to Micro Focus
browser.DomElement("//div[@id='mylinks']").GetProperty("innerHTML")	This is my link collection: Bye bye Borland Welcome to Micro Focus

 **注:** Silk Test 13.5 以降で、要素の textContents プロパティを介して取得されるテキストの空白はサポートするブラウザすべてで一貫してトリムされます。ブラウザのバージョンによっては、この空白の処理方法が Silk Test 13.5 以前の Silk Test バージョンで異なる場合があります。以前の動作を有

効にしたい場合、OPT_COMPATIBILITY オプションを 13.5.0 より低いバージョンに設定してください。たとえば、オプションを Silk Test 13.0 に設定するには、スクリプトで次のように入力します：

```
'VB .NET code
Agent.SetOption("OPT_COMPATIBILITY", "13.0.0")
```

クロス ブラウザー スクリプトの作成時に必要な処置

クロス ブラウザー スクリプトを作成する場合は、以下の 1 つまたは複数の問題に遭遇する場合があります。

- クロス ブラウザー テスト用にスクリプトを記録する場合、Google Chrome、Mozilla Firefox、または Microsoft Edge を使用することを Micro Focus では推奨しています。Internet Explorer を使って Silk Test Workbench で記録したスクリプトは、ほかのブラウザで記録したスクリプトと若干異なる場合があります。
- 属性値が異なる。たとえば、Internet Explorer の色が "# FF0000" として、Mozilla Firefox の色が "rgb(255,0,0)" として返されます。
- 属性名が異なる。たとえば、Internet Explorer 8 以前のバージョンのフォント サイズ属性は "fontSize" でしたが、Internet Explorer 9 以降および Mozilla Firefox などの他のすべてのブラウザでは "font-size" となります。
- 一部のフレームワークで異なる DOM ツリーがレンダリングされることがある

安定したクロス ブラウザ テストを実現するために最適なロケーター

組み込みロケーター生成プログラムでは、安定したロケーターの作成が試みられます。ただし、情報を使用できない場合、高品質のロケーターを生成することは困難です。この場合、ロケーター生成プログラムでは、階層形式の情報およびインデックスが使用されます。その結果、直接的な記録/再生には適していても、安定した日常的な実行には適さない脆弱なロケーターが生成されます。さらに、クロス ブラウザ テストでは、いくつかの AJAX フレームワークで異なるブラウザに対して異なる DOM 階層がレンダリングされることがあります。

この問題を回避するには、アプリケーションの UI 要素にカスタム ID を使用します。

ロケーターでクラスとスタイルの属性が使用されない理由

これらの属性は AJAX アプリケーションで頻繁に変更され、ロケーターの安定性が損なわれることがあり、ため、無視リストに含まれています。ただし、多くの場合、これらの属性を使用してオブジェクトを識別できるため、アプリケーションで使用することに意味がある場合があります。

Internet Explorer 10 で Click の記録が異なる理由

Internet Explorer 10 の DomElement で Click を記録し、DomElement が Click の後で破棄された場合、記録動作が予期したとおりにならないことがあります。別の DomElement が最初の DomElement の下にある場合、Silk Test では、1 つの Click が記録されるのではなく、Click、MouseMove、および ReleaseMouse が記録されます。

予期しない記録動作を回避する方法は、テスト対象のアプリケーションによって異なります。通常は、記録されたスクリプトから不必要な MouseMove イベントと ReleaseMouse イベントを削除すれば十分です。

ハンドル無効エラーが表示される理由

このトピックでは、Silk Test Workbench に次のエラー メッセージが表示された場合の対処法について説明します：「このオブジェクトのハンドルは無効になりました。」

このメッセージは、たとえば Click などのメソッドを呼び出したオブジェクトが何らかの理由で消失していることを示しています。たとえば、Web アプリケーションでメソッドを呼び出しているときに、何らかの理由でブラウザが新しいページに移動した場合、以前のページのすべてのオブジェクトは自動的に無効になります。

Web アプリケーションのテストでは、組み込みの同期がこの問題の原因の場合があります。たとえば、テスト対象のアプリケーションにショッピング カートが含まれていて、このショッピング カートに品物を追加したとします。ユーザーは次のページが読み込まれ、ショッピング カートのステータスが 品物がある 状態に変わるまで待機しています。品物を追加するという操作からの戻り時間が短すぎた場合、最初のページのショッピング カートはステータスが変わるまで待機しますが、その間も新しいページは読み込まれています。したがって、最初のページのショッピング カートは無効になります。この動作によって、ハンドル無効エラーが発生します。

この問題を回避するには、2 番目のページでのみ有効なオブジェクトが表示されるまで待機してから、ショッピング カートのステータスを確認するようにしてください。このオブジェクトが有効になるとすぐに、ショッピング カートのステータスを確認できるようになり、2 番目のページで正しく検証されるようになります。

スクリプトからのブラウザの起動

テストの開始時に再生するブラウザを選択するのではなく、再生時にテスト スクリプトから特定のブラウザを起動することが必要になる場合があります。

BrowserBaseState クラスの使用

テスト スクリプトからブラウザを起動するために、BrowserBaseState クラスを使用すると、Executable プロパティで指定したブラウザが実行され、テストの準備ができています。さらに、基本状態として Url プロパティで指定した URL に移動され、ブラウザが最前面に表示されます。

次のサンプル コードは、BrowserBaseState を使用して Internet Explorer を起動します。

```
' VB .NET code
Dim MyBrowserBaseState As New BrowserBaseState(BrowserType.InternetExplorer, "http://
www.borland.com")
MyBrowserBaseState.Execute()
```

ブラウザの複数インスタンスの使用

複数のブラウザ ウィンドウまたはタブが開かれている場合、Silk Test Workbench は各ブラウザ ウィンドウまたはタブを、ユニークなロケーターを持つ個別のオブジェクトとして処理します。ロケーターは、WebBrowser、WebBrowser[1]、WebBrowser[2] のように、インデックス付きで表されます。

非表示入力フィールドの検索

非表示入力フィールドは、タグに type="hidden" を指定した HTML フィールドです。Find で非表示入力フィールドを検索できるようにするために、OPT_XBROWSER_FIND_HIDDEN_INPUT_FIELDS オプションを使用できます。このオプションのデフォルト値は、TRUE です。

```
'VB .NET code
Agent.SetOption("OPT_XBROWSER_FIND_HIDDEN_INPUT_FIELDS", true)
```

```
// C# code
Agent.SetOption("OPT_XBROWSER_FIND_HIDDEN_INPUT_FIELDS", true);
```

Web アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジ ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Web アプリケーションがサポートする属性は次のとおりです。

- caption (次のワイルドカードをサポート : ? および *)
- すべての DOM 属性 (次のワイルドカードをサポート : ? および *)



注: 各ブラウザによって、空のスペースの処理に違いがあります。この結果、「textContent」および「innerText」属性は正規化されています。空のスペースのあとに別の空のスペースが続く場合、空のスペースはスキップされるか、または 1 文字の空白で置き換えられます。空のスペースとは、検出されたスペース、キャリッジリターン、改行、タブのことです。また、このような値に一致するものも正規化されます。例：

```
<a>abc  
abc</a>
```

以下のロケータを使用します。

```
//A[@innerText='abc abc']
```

Web アプリケーションのカスタム属性

HTML は、安定した識別子を表すことができる一般的な属性 ID を定義します。定義により、ID は文書内の要素を一意に識別します。特定の ID を持つ要素は文書内で 1 つだけ存在します。

ただし、多くの場合 (特に AJAX アプリケーションでは)、ID は HTML 要素に関連付けられたサーバー ハンドラを動的に識別するために使用されます。つまり、Web 文書の作成のたびに ID は変わることになります。このような場合、ID は安定した識別子ではなく、Web アプリケーションの UI コントロールを識別するのに適しません。

Web アプリケーションの場合、より確実にするには、Silk Test Workbench に UI コントロールの情報を公開するためだけに使用されるカスタム HTML 属性を新たに導入することです。

カスタム HTML 属性はブラウザは無視するため、AUT の動作は変わりません。ブラウザの DOM を通じてアクセスすることができます。Silk Test Workbench では、このような属性を (属性がコントロール クラスのカスタム 属性であっても) 識別時のデフォルト属性として使用するように設定することができます。特定のテクノロジ ドメインのデフォルト識別属性としてカスタム属性を設定するには、**ツール > オプション > 記録** をクリックして、テクノロジ ドメインを選択します。

アプリケーション開発者は、Web 要素にさらに HTML 属性を追加することが必要です。

元の HTML コード：

```
<A HREF="http://abc.com/control=4543772788784322..." <IMG  
src="http://abc.com/xxx.gif" width=16 height=16> </A>
```

新しいカスタム HTML 属性 *AUTOMATION_ID* を持つ HTML コード：

```
<A HREF="http://abc.com/control=4543772788784322..."  
AUTOMATION_ID = "AID_Login" <IMG src="http://abc.com/xxx.gif"  
width=16 height=16> </A>
```

カスタム属性を設定すると、Silk Test Workbench は、できる限りカスタム属性を使用して、一意のロケータを構成しようとします。Web ロケータは次のようになります。

```
...//DomLink[@AUTOMATION_ID='AID_Login']
```

例：変化する ID

変化する ID の 1 例は、Google Widget Toolkit (GWT) で、ID は Web 文書の作成のたびに変化する動的な値を保持します：

```
ID = 'gwt-uid-<nnn>'
```

この場合、<nnn> が頻繁に変化します。

xBrowser クラス リファレンス

xBrowser アプリケーションを設定すると、Silk Test Workbench は標準の xBrowser コントロールのテストのサポートを組み込みで提供します。

Microsoft Windows 8 および Microsoft Windows 8.1 上のテストの制限事項

以下に、Microsoft Windows 8 および Microsoft Windows 8.1 上でテストする際の既知の制限事項を一覧します。

- Silk Test は、Microsoft Windows 8 および Microsoft Windows 8.1 上でのユニバーサル Windows プラットフォーム (UWP) アプリ (Windows Store アプリ、または Metro スタイル アプリともいう) のテストをサポートしません。

サポートする属性の種類

ロケーターが作成される時、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。必要に応じて、以下のいずれかの方法を使用して属性の種類を変更できます。

- 他の属性の種類と値を手動で入力する。
- **推奨属性リスト** の値を変更して、デフォルトの属性の種類に対して別の設定を指定する。

Apache Flex アプリケーションの属性

ロケーターが作成される時、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Flex アプリケーションがサポートする属性は次のとおりです。

- automationName
- caption (automationName と同様)
- automationClassName (FlexButton など)
- className (実装クラスの完全修飾名。 mx.controls.Button など)
- automationIndex (FlexAutomation のビューでのコントロールのインデックス。 index:1 など)
- index (automationIndex と同様。ただし、接頭辞はなし。 1 など)
- id (コントロールの ID)
- windowId (id と同様)
- label (コントロールのラベル)
- すべての動的ロケーター属性



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

動的ロケーター属性の詳細については、「動的ロケーター属性」を参照してください。

Java AWT/Swing アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Java AWT/Swing でサポートされる属性には以下のものがあります。

- caption
- priorlabel : 隣接するラベル フィールドのテキストによってテキスト入力フィールドを識別します。通常、フォームのすべての入力フィールドに、入力の目的を説明するラベルがあります。caption のないコントロールの場合、自動的に属性 **priorlabel** がロケーターに使用されます。コントロールの **priorlabel** 値 (テキスト入力フィールドなど) には、コントロールの左側または上にある最も近いラベルの caption が使用されます。
- name
- accessibleName
- *Swing* のみ : すべてのカスタム オブジェクトの定義属性は、ウィジェットに `putClientProperty("propertyName", "propertyValue")` で設定されます。



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

Java SWT アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

Java SWT がサポートする属性は次のとおりです。

- caption
- すべてのカスタム オブジェクト定義属性



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

SAP アプリケーションの属性

ロケーターが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケーターがテスト内のオブジェクトを識別する方法が決定されます。

SAP がサポートする属性は次のとおりです。

- automationId
- caption



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケーター属性は、ワイルドカード ? および * をサポートしています。

Silverlight コントロールを識別するためのロケータ属性

Silverlight コントロールでサポートされているロケータ属性は次のとおりです。

- *automationId*
- *caption*
- *className*
- *name*
- すべての動的ロケータ属性



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

Silverlight スクリプト内のコンポーネントを識別するために、*automationId*、*caption*、*className*、*name*、または任意の動的ロケータ属性を指定できます。*automationId* はアプリケーション開発者が設定します。たとえば、*automationId* を持つロケータは、以下のようになります：
`// SLButton[@automationId="okButton"]`

automationId は一般に非常に有用で安定した属性であるため、使用することを推奨します。

属性の種類	説明	例
<i>automationId</i>	テスト対象アプリケーションの開発者によって設定される識別子。Visual Studio デザイナは、デザイナー上で作成されたすべてのコントロールに自動的に <i>automationId</i> を割り当てます。アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この ID を使用します。	<code>// SLButton[@automationId="okButton"]</code>
<i>caption</i>	コントロールが表示するテキスト。複数の言語にローカライズされたアプリケーションをテストする場合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	<code>//SLButton[@caption="Ok"]</code>
<i>className</i>	Silverlight コントロールの .NET 単純クラス名 (名前空間なし)。 <i>className</i> 属性を使用すると、Silk Test Workbench が解決する標準 Silverlight コントロールから派生したカスタム コントロールを識別するのに役立ちます。	<code>// SLButton[@className='MyCustomButton']</code>
<i>name</i>	コントロールの名前。テスト対象アプリケーションの開発者によって設定されます。	<code>//SLButton[@name="okButton"]</code>



注目: XAML コードの *name* 属性は、ロケータ属性 *name* ではなく、ロケータ属性 *automationId* にマップされます。

Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性をこの表に示した順番に使用して Silverlight コントロールのロケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、*automationId* が固有の場合は Silk Test Workbench がロケータを作成する際に使用されます。

以下の表は、アプリケーション開発者がテキスト「Ok」を持つ Silverlight ボタンをアプリケーションの XAML コードに定義する方法を示しています。

オブジェクトの XAML コード	Silk Test からオブジェクトを検索するためのロケータ
<Button>Ok</Button>	//SLButton[@caption="Ok"]
<Button Name="okButton">Ok</Button>	//SLButton[@automationId="okButton"]
<Button AutomationProperties.AutomationId="okButton">Ok</Button>	//SLButton[@automationId="okButton"]
<Button AutomationProperties.Name="okButton">Ok</Button>	//SLButton[@name="okButton"]

UI オートメーションを使ってコントロールを識別するためのロケータ属性

UI オートメーション プロバイダー インターフェイスを実装した Windows ベースのアプリケーションのコントロールに対して、次のロケータ属性がサポートされています。

- *automationId*
- *caption*
- *className*
- *name*
- すべての動的ロケータ属性



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

UI オートメーション プロバイダー インターフェイスを実装した Windows ベースのアプリケーションのコントロールを識別するには、スクリプトで *automationId*、*caption*、*className*、*name*、またはすべての動的ロケータ属性を使用できます。*automationId* はアプリケーション開発者が設定します。たとえば、*automationId* を持つロケータは、以下のようになります: //

```
UIAButton[@automationId="okButton"]
```

automationId は一般に非常に有用で安定した属性であるため、使用することを推奨します。

属性の種類	説明	例
<i>automationId</i>	テスト対象アプリケーションの開発者によって設定される識別子。Visual Studio デザイナは、デザイナー上で作成されたすべてのコントロールに自動的に <i>automationId</i> を割り当てます。アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この ID を使用します。	// UIAButton[@automationId="okButton"]
<i>caption</i>	コントロールが表示するテキスト。複数の言語にローカライズされたアプリケーションをテストする場合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	//UIAButton[@caption="Ok"]
<i>className</i>	UI オートメーション コントロールのクラス名 (名前空間なし)。 <i>className</i> 属性を使用すると、Silk Test Workbench が解決する標準 UI オートメーション コントロールから派生したカスタム コントロールを識別するのに役立ちます。	// UIAButton[@className='MyCustomButton']

属性の種類	説明	例
name	コントロールの名前。テスト対象アプリケーションの開発者によって設定されます。	//UIButton[@name="okButton"]


Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性を、この表に示した順番に使用して UI オートメーション コントロールのロケータを記録時に作成します。たとえば、コントロールが *automationId* と *name* を持つ場合、*automationId* を使って固有に識別できれば、Silk Test Workbench によるロケータの作成に使用されます。

ユーザーの AUT の UI オートメーション コントロールに対して使用できるその他のカスタム属性を確認するには、**プロパティの検証** ダイアログ ボックスを使用できます。使用するときは、記録する UI オートメーション コントロール上にマウス カーソルを移動し、**Ctrl + Alt** をクリックします。これで、コントロールに対して使用できるプロパティを確認できます。たとえば、アプリケーションによっては、属性 value が有効な場合もあります。

Rumba コントロールを識別するためのロケータ属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。サポートされている属性は次のとおりです。

caption	コントロールが表示するテキスト。
priorlabel	フォームの入力フィールドには通常入力の目的を説明するラベルがあるため、 priorlabel の目的は隣接するラベル フィールド RumbaLabel のテキストによってテキスト入力フィールド RumbaTextField を識別することです。テキスト フィールドの同じ行の直前にラベルがない場合、または右側のラベルが左側のラベルよりテキスト フィールドに近い場合、テキスト フィールドの右側にあるラベルが使用されます。
StartRow	この属性は記録されていませんが、手動でロケータに追加することができます。 StartRow を使用して、この行で始まるテキスト入力フィールド、 RumbaTextField を識別します。
StartColumn	この属性は記録されていませんが、手動でロケータに追加することができます。 StartColumn を使用して、この列で始まるテキスト入力フィールド、 RumbaTextField を識別します。
すべての動的ロケータ属性。	動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

Web アプリケーションの属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。

Web アプリケーションがサポートする属性は次のとおりです。

- caption (次のワイルドカードをサポート : ? および *)
- すべての DOM 属性 (次のワイルドカードをサポート : ? および *)



注: 各ブラウザによって、空のスペースの処理に違いがあります。この結果、「textContent」および「innerText」属性は正規化されています。空のスペースのあとに別の空のスペースが続く場合、空のスペースはスキップされるか、または 1 文字の空白で置き換えられます。空のスペースとは、検出されたスペース、キャリッジリターン、改行、タブのことです。また、このような値に一致するものも正規化されます。例：

```
<a>abc  
abc</a>
```

以下のロケータを使用します。

```
//A[@innerText='abc abc']
```

Windows Forms アプリケーションの属性

ロケータが作成される時、属性の種類はアプリケーションが使用するテクノロジドメインに基づいて自動的に割り当てられます。属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。

Windows Forms アプリケーションがサポートする属性は次のとおりです。

- automationid
- caption
- windowid
- priorlabel (caption のないコントロールの場合、自動的に priorlabel が caption として使用されます。caption のあるコントロールの場合、caption を使う方が簡単な場合があります。)



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

Windows Presentation Foundation (WPF) アプリケーションの属性

WPF アプリケーションがサポートする属性は次のとおりです。

- automationId
- caption
- className
- name
- すべての動的ロケータ属性。



注: 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性の詳細については、「動的ロケータ属性」を参照してください。

オブジェクト解決

WPF スクリプト内のコンポーネントを識別するために、automationId、caption、className、あるいは name を指定できます。アプリケーション中の要素に指定された name が利用可能な場合、ロケータの automationId 属性として使用されます。この結果、多くのオブジェクトは、この属性のみを使用して一意に識別できます。たとえば、automationId を持つロケータは、以下のようになります：//

```
WPFButton[@automationId='okButton']"
```

automationId や他の属性を定義した場合、再生中に automationId だけが使用されます。automationId が定義されていない場合には、コンポーネントを解決するのに name が使用されます。name も

automationId もどちらも定義されていない場合には、*caption* 値が使用されます。 *caption* が定義されていない場合は、*className* が使用されます。 *automationId* は非常に役立つプロパティであるため、使用することを推奨します。

属性の種類	説明	例
<i>automationId</i>	テスト アプリケーションの開発者によって提供された ID	//WPFButton[@automationId='okButton']"
<i>name</i>	コントロールの名前。 Visual Studio デザイナは、デザイナー上で作成されたすべてのコントロールに自動的に名前を割り当てます。 アプリケーション開発者は、アプリケーションのコード上でコントロールを識別するために、この名前を使用します。	//WPFButton[@name='okButton']"
<i>caption</i>	コントロールが表示するテキスト。 複数の言語にローカライズされたアプリケーションをテストする場合、 <i>caption</i> の代わりに <i>automationId</i> や <i>name</i> 属性を使用することを推奨します。	//WPFButton[@automationId='Ok']"
<i>className</i>	WPF の .NET 単純クラス名 (名前空間なし)。 クラス名属性を使用すると、Silk Test Workbench が解決する標準 WPF コントロールから派生したカスタム コントロールを識別するのに役立ちます。	//WPFButton[@className='MyCustomButton']"

Silk Test Workbench は、*automationId*、*name*、*caption*、*className* 属性をこの表に示した順番に使用して WPF コントロールのロケータを記録時に作成します。 たとえば、コントロールが *automationId* と *name* を持つ場合、Silk Test Workbench がロケータを作成する際には *automationId* が使用されます。

以下の例では、アプリケーション開発者がアプリケーションの WPF ボタンに対して *name* と *automationId* を XAML コードに定義する方法を示します。


```
<Button Name="okButton" AutomationProperties.AutomationId="okButton"
Click="okButton_Click">Ok</Button>
```

Windows API ベースのクライアント/サーバー アプリケーションの属性

ロケータが作成されるとき、属性の種類はアプリケーションが使用するテクノロジー ドメインに基づいて自動的に割り当てられます。 属性の種類と値によって、ロケータがテスト内のオブジェクトを識別する方法が決定されます。


Windows API ベースのクライアント/サーバー アプリケーションがサポートする属性は次のとおりです。

- caption
- windowid
- priorlabel : 隣接するラベル フィールドのテキストによってテキスト入力フィールドを識別します。通常、フォームのすべての入力フィールドに、入力の目的を説明するラベルがあります。caption のないコントロールの場合、自動的に属性 **priorlabel** がロケータに使用されます。コントロールの **priorlabel** 値 (テキスト ボックスなど) には、コントロールの左側または上にある最も近いラベルの caption が使用されます。

 **注:** 属性名は、大文字小文字が区別されます (モバイル アプリケーションを除く。モバイル アプリケーションでは、大文字小文字は無視されます)。デフォルトで、属性値では大文字と小文字が区別されますが、他のオプションと同様にこのデフォルト設定は変更できます。ロケータ属性は、ワイルドカード ? および * をサポートしています。

動的ロケータ属性

再生中にコントロールを識別するために、事前に定義されたロケータ属性のセット (*caption* や *automationId* など。テクノロジ ドメインに依存します) をロケータに使用できます。しかし、動的プロパティを含む、コントロールのすべての属性をロケータ属性として使用することもできます。特定のコントロールで使用可能なプロパティのリストを取得するには、GetPropertyList メソッドを使用します。返されたプロパティはすべて、ロケータを使用してコントロールを識別するのに使用できます。

 **注:** 特定のプロパティの実際の値を取得するには、GetProperty メソッドを使用します。この値はロケータで使用できます。

例

Silverlight アプリケーションのダイアログ ボックスにあるボタンを識別する場合、以下のように入力します。

```
browser.Find("//SLButton[@IsKeyboardFocused=true]")
```

または

```
Dim button = dialog.SLButton("@IsKeyboardFocused=true")
```

これが機能するのは、Silk Test Workbench により Silverlight ボタン コントロールの IsDefault というプロパティが公開されるためです。

例

Silverlight アプリケーションのフォント サイズ 12 のボタンを識別する場合、以下のように入力します。

```
Dim button = browser.Find("//SLButton[@FontSize=12]")
```

または

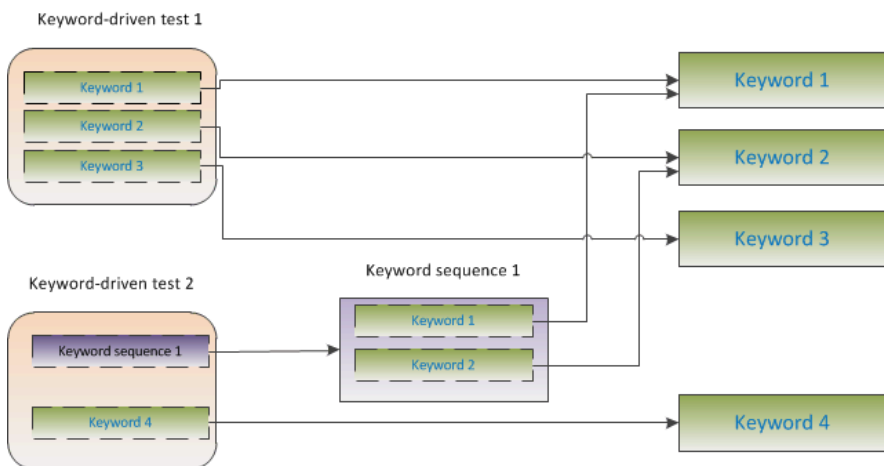
```
Dim button = browser.SLButton("@FontSize=12")
```

これが機能するのは、テスト対象アプリケーションの基になるコントロール (この場合、Silverlight ボタン) が FontSize というプロパティを持つためです。

キーワード駆動テスト

キーワード駆動テストは、テスト開発からテスト設計を分離するソフトウェアテスト手法です。このため、テスト自動化プロセスにビジネスアナリストなどの専門家を含めることができます。Silk Central と Silk Test はキーワード駆動テストをサポートしており、Silk Test のキーワードの形式での共有資産として構成されるメンテナンス可能な自動化フレームワークを自動化エンジニアが開発することによって、自動化エンジニアとビジネスアナリスト間で密接な共同作業を行うことができます。その後、これらのキーワードは、Silk Test で新しいキーワード駆動テストを作成したり、Silk Central で既存の手動テスト資産を自動テストに変換したり、新しいキーワード駆動テストを作成するために、ビジネスアナリストが使用することができます。

- キーワード駆動テストは、実行可能なキーワードのコレクションです。キーワード駆動テストは、他のテストと同様に再生することができます。
- キーワードシーケンスは、他のキーワードを組み合わせたものです。キーワードシーケンスは、頻繁に使用するキーワードの組み合わせを1つのキーワードにまとめることにより、メンテナンスの労力を低減し、テストを理解しやすくすることができます。
- キーワードは、テストオブジェクトに対する複数の操作の組み合わせを定義したものです。キーワードの実装は、さまざまなツールとプログラム言語 (Java や .NET など) を使用して行えます。



キーワード駆動テストの作成には、次の2つのフェーズがあります。

1. テストの設計
2. キーワードの実装

キーワード駆動テストで利用可能な記録/再生コントロールの完全な一覧については、「キーワード駆動テストクラスリファレンス」を参照してください。

キーワード駆動テストの利点

キーワード駆動テスト手法を使用する利点を次に示します。

- キーワード駆動テストを使用すると、テスト自動化とテストケースのデザインが分離され、うまく分業できるようになり、キーワードを実装するテストエンジニアとテストケースをデザインする専門家が共同作業できます。
- テスト対象アプリケーションにアクセスすることなく、初期の段階からテストを開発でき、後からキーワードを実装できます。
- プログラムの知識がなくてもテストを開発できます。

- キーワード駆動テストは、長期的に見るとメンテナンス コストを低減できます。キーワードのメンテナンスが必要で、これらのキーワードを使用するすべてのキーワード駆動テストは自動的に更新されます。
- テストケースが簡潔です。
- 技術者でなくてもテスト ケースが読みやすく、理解しやすくなります。
- テスト ケースの変更が簡単です。
- 既存のキーワードを再利用して新しいテストを再利用できます。これにより、より広範囲なテスト カバレッジを実現しやすくなります。
- キーワード実装の内部的な複雑性を、キーワード駆動テストを作成または実行するユーザーに対して隠蔽できます。

キーワード

キーワードは、テスト オブジェクトに対する複数の操作の組み合わせを定義したものです。キーワードの実装は、さまざまなツールとプログラム言語 (Java や .NET など) を使用して行えます。Silk Test Workbench でのキーワードは、ビジュアル テストまたはメソッド名の前に Keyword 属性を持つメソッドです。キーワードは、キーワード資産として保存されます。

キーワード駆動テストの作成中にキーワードやキーワード シーケンスを定義し、後でそれらをテスト メソッド、またはビジュアル テストとして実装できます。既存のテスト メソッドまたはビジュアル テストに Keyword 属性を付けて、キーワードとしてマークすることもできます。キーワードは次のように定義されます：

```
'VB .NET code
<Keyword("keyword_name")>
```

キーワード シーケンスは、他のキーワードを組み合わせたものです。キーワード シーケンスは、頻繁に使用するキーワードの組み合わせを 1 つのキーワードにまとめることにより、メンテナンスの労力を低減し、テストを理解しやすくすることができます。

キーワードまたはキーワード・シーケンスは、合計 20 の入力および出力パラメーターを持つことができます。キーワードを実装するテスト メソッドやビジュアル テストのパラメータは、キーワードのパラメータです。キーワードのパラメータに違う名前を指定するために、次を使用できます：

```
'VB .NET code
Argument("parameter_name")
```

例

キーワードとしてマークされたテスト メソッドは、次のようになります。

```
'VB .NET code
<Keyword("Login")>
Public Sub Login()
    ... // method implementation
End Sub
```

または

```
'VB .NET code
<Keyword("Login", Description:="Logs in with the given name and
password.")>
Public Sub Login(<Argument("UserName")> username As String,
<Argument("Password")> password As String)
    ... // method implementation
End Sub
```

このキーワードは、指定したユーザー名とパスワードを使ってテスト対象アプリケーションにログインします。



注: ヘルプ トピックを PDF で参照している場合、このサンプルコードは、実際のスクリプトでは許されない場所で改行されてしまっている場合があります。スクリプトでこのサンプルコードを使用する場合は、これらの改行を削除してください。

- Keyword 属性のキーワード名パラメータは、省略可能です。メソッドの名前とは異なる名前を指定する場合に、キーワード名パラメータを使用できます。パラメータが指定されていない場合、メソッドの名前がキーワード名として使用されます。
- Argument 属性も省略可能です。メソッドをキーワードとしてマークすると、自動的にすべての引数がキーワードの引数として使用されます。例のように、`userName` を `UserName` にしたい場合など、キーワードの引数とは異なる名前を指定する場合に、Argument 属性を使用できます。

Silk Test Workbench でキーワード駆動テストを作成する

キーワード駆動テスト エディター を使って、新しいキーワードと既存のキーワードを新しいキーワード駆動テストに結合できます。新しいキーワードは、後のステップでビジュアル テストまたは VB .NET スクリプトのメソッドとして実装する必要があります。

1. **ファイル** > **新規作成** を選択します。 **資産の新規作成** ダイアログ ボックスが開きます。
2. 資産の種類のリストから **キーワード駆動テスト** を選択します。
3. 新しい資産の名前を **資産名** フィールドに入力します。
4. 新しい資産を追加したいプロジェクトを選択します。

デフォルトでは、プロジェクトがアクティブであれば、そのアクティブなプロジェクトに新しい資産が作成されます。



注: Silk Test Workbench が提供する機能を最適に使用するには、同じテストで複数のアプリケーションをテストする場合を除き、テストするアプリケーションごとに個別のプロジェクトを作成します。

5. **OK** をクリックします。 **キーワード駆動テスト エディター** が開きます。
6. 次のアクションのいずれかを実行します。
 - 新しいキーワードを追加する場合は、 **新しいキーワード** フィールドにキーワードの名前を入力します。
 - 既存のキーワードを追加する場合は、 リストを展開して追加するキーワードを選択します。
7. Enter を押します。
8. 実行するすべてのキーワードを追加するまで、上記の 2 つの手順を繰り返します。
9. **保存** をクリックします。

続いて、キーワードを実装します。すべてのキーワードが実装されている場合は、テストを実行します。


Silk Test Workbench でのキーワード駆動テストの記録

単一のキーワードを記録する場合は、「[キーワードの記録](#)」を参照してください。


キーワード駆動テストを記録するには：

1. **ファイル** > **新規作成** を選択します。 **資産の新規作成** ダイアログ ボックスが開きます。
2. 資産の種類のリストから **キーワード駆動テスト** を選択します。
3. 新しい資産の名前を **資産名** フィールドに入力します。
4. 新しい資産を追加したいプロジェクトを選択します。

デフォルトでは、プロジェクトがアクティブであれば、そのアクティブなプロジェクトに新しい資産が作成されます。

 **注:** Silk Test Workbench が提供する機能を最適に使用するには、同じテストで複数のアプリケーションをテストする場合を除き、テストするアプリケーションごとに個別のプロジェクトを作成します。

5. 記録をすぐに開始するために、**記録の開始** チェック ボックスをオンにします。
6. **OK** をクリックします。
7. **記録対象** ダイアログ ボックスで、次のアクションを実行します。
 - a) **資産の種類** フィールドで、キーワード駆動テストを .NET スクリプトとして保存するか、ビジュアルテストとして保存するかを選択します。
 - b) 省略可能：キーワード駆動テストを .NET スクリプトとして保存するように選択した場合、**資産名** フィールドに .NET スクリプトの名前を指定します。
デフォルトの名前は、最後に使用した .NET スクリプトの名前です。
 - c) **OK** をクリックします。
8. 現在のプロジェクトに対してアプリケーション構成が設定されており、Web アプリケーションをテストする場合、**ブラウザーの選択** ダイアログ ボックスが開きます。
 - a) ブラウザーを選択します。
 - b) 省略可能：あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザー ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザー サイズの編集** をクリックすると、新しいブラウザー サイズを指定したり、**ブラウザー サイズ** リストに表示するブラウザー サイズを選択することができます。
9. 開いているダイアログに応じて、次のいずれかを実行します。
 - **アプリケーションの選択** ダイアログ ボックスで、**OK** をクリックします。
 - **ブラウザーの選択** ダイアログ ボックスで、**記録** をクリックします。
10. テスト対象アプリケーションで、最初のキーワードに含める操作を実行します。
記録中に利用可能な操作についての詳細は、「記録中に利用可能な操作」を参照してください。
11. キーワードの名前を指定するには、**記録中** ウィンドウでキーワードの名前の上にマウスカーソルを動かして、**編集** をクリックします。

 **注:** Silk Test Workbench は、キーワード駆動テストの開始にアプリケーションの開始 キーワードを自動的に追加します。このキーワードで、アプリケーションの基本状態が実行され、テストを正しく再生できるようになります。基本状態についての詳細は、「基本状態」を参照してください。
12. キーワードの名前を **キーワードの名前** フィールドに入力します。
13. **OK** をクリックします。
14. 次のキーワードの操作を記録するには、**新しいキーワードの名前** フィールドに新しいキーワードの名前を入力し、**追加** をクリックします。Silk Test Workbench は、新しいキーワードに新しい操作を記録します。
15. キーワード駆動テスト全体を記録すまで、新しいキーワードを作成し、キーワードに対する操作を記録します。
16. **停止** をクリックします。 **記録完了** ダイアログ ボックスが開きます。

Silk Test Workbench は、すべての記録したキーワードを含む新しいキーワード駆動テストを作成します。

Silk Test Workbench でのキーワード駆動テストの基本状態の設定

Silk Test Workbench でキーワード駆動テストを実行すると、キーワード駆動テストは基本状態のキーワードを呼び出すことにより、Silk Test Workbench は AUT を基本状態から開始します。

キーワード駆動テストの記録時に、Silk Test Workbench は基本状態のキーワード (IsBaseState プロパティが *True* に設定されているキーワード) を現在のプロジェクトから検索します。

- 基本状態のキーワードが現在のプロジェクトに存在した場合、Silk Test Workbench は、キーワード駆動テストの最初のキーワードとして、このキーワードを挿入します。
- 基本状態のキーワードがプロジェクトに存在しない場合、Silk Test Workbench は、アプリケーションの開始 という名前の新しい基本状態のキーワードを作成し、キーワード駆動テストの最初のキーワードとして挿入します。

.NET メソッドとして実装されたキーワードを手動で基本状態のキーワードとしてマークするには、IsBaseState プロパティを Keyword 属性に追加し、プロパティの値を *True* に設定します。

```
<Keyword( "Start application" , IsBaseState:= True )>  
Public Sub Start_application()  
End Sub
```

ビジュアルテストとして実装されたキーワードを手動で基本状態のキーワードとしてマークするには、ビジュアルテストの <<開始>> ステップのプロパティを開き、**キーワード** セクションの **基本状態** プロパティを **はい** に設定します。

キーワード駆動テストのアプリケーション構成は、現在のプロジェクトに保存されます。アプリケーション構成を管理するには、Silk Test Workbench のメニューから **ツール** > **アプリケーション構成の編集** を選択します。これによって、**プロジェクトの選択** ダイアログ ボックスが開き、アクセス可能なすべてのプロジェクトが表示されます。プロジェクトを選択し、**編集** をクリックしてアプリケーション構成を編集します。

基本状態を含むキーワード駆動テストを実行すると、そのキーワード駆動テストが定義されているプロジェクトのアプリケーション構成が使用されます。

Silk Test Workbench でのキーワードの実装

キーワードを実装する前に、キーワード駆動テストの一部としてキーワードを定義します。

キーワード駆動テストで再利用するためにキーワードを実装するには：

1. 実装するキーワードが含まれているキーワード駆動テストを開きます。
2. **キーワード駆動テスト エディター** で、実装するキーワードの左側に表示される **キーワードの実装** をクリックします。 **キーワードの実装** ダイアログ ボックスが開きます。
3. 新しいキーワードの資産の種類を選択します。
 - .NET メソッドとしてキーワードを実装するには、**.NET スクリプト** をクリックします。既存の .NET スクリプトに新しいキーワードを追加したり、**資産名** フィールドに名前を入力して、キーワードに対して新しい .NET スクリプトを作成できます。
 - キーワードシーケンスとしてキーワードを実装するには、**キーワード** をクリックします。中間レベルのキーワードは、他のキーワードのシーケンスであるため、記録することはできません。
 - ビジュアルテストとしてキーワードを実装するには、**ビジュアルテスト** をクリックします。
4. キーワードの名前を **資産名** フィールドに入力します。
5. 省略可能：キーワードを記録するには、**記録の開始** チェック ボックスをオンにします。

6. **OK** をクリックします。
7. **記録** をクリックします。

記録の詳細については、「[キーワードの記録](#)」を参照してください。

Silk Test Workbench でのキーワードの記録

完全に新しいキーワードに対してではなく、キーワード駆動テストに既に存在するキーワードに対しては、操作の記録のみを行えます。新しいキーワード駆動テストを記録する場合は、「[キーワード駆動テストの記録](#)」を参照してください。

新しいキーワードの操作を記録するには：

1. 記録するキーワードが含まれているキーワード駆動テストを開きます。
2. **キーワード駆動テスト エディター** で、実装するキーワードの左側に表示される **キーワードの実装** をクリックします。 **キーワードの実装** ダイアログ ボックスが開きます。
3. 新しいキーワードの資産の種類を選択します。
 - .NET メソッドとしてキーワードを実装するには、**.NET スクリプト** をクリックします。既存の .NET スクリプトに新しいキーワードを追加したり、**資産名** フィールドに名前を入力して、キーワードに対して新しい .NET スクリプトを作成できます。
 - キーワード シーケンスとしてキーワードを実装するには、**キーワード** をクリックします。中間レベルのキーワードは、他のキーワードのシーケンスであるため、記録することはできません。
 - ビジュアルテストとしてキーワードを実装するには、**ビジュアルテスト** をクリックします。
4. **記録の開始** チェック ボックスをチェックします。
5. **OK** をクリックします。
6. 現在のプロジェクトに対してアプリケーション構成が設定されており、Web アプリケーションをテストする場合、**ブラウザーの選択** ダイアログ ボックスが開きます。
 - a) ブラウザーを選択します。
 - b) 省略可能：あらかじめ定義されたブラウザー サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザー サイズ** リストからブラウザー サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザー ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザー ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザー サイズの編集** をクリックすると、新しいブラウザー サイズを指定したり、**ブラウザー サイズ** リストに表示するブラウザー サイズを選択することができます。
7. テスト対象アプリケーションで、テストする操作を実行します。
記録中に利用可能な操作についての詳細は、「[記録中に利用可能な操作](#)」を参照してください。
8. **停止** をクリックします。 **記録完了** ダイアログ ボックスが開きます。

記録した操作は、定義したビジュアルテストまたは .NET スクリプトに表示されます。

ビジュアルテストをキーワードとして指定

既存のビジュアルテストをキーワードとして指定して、キーワード駆動テストで使用することができます。

1. キーワードとして指定するビジュアルテストを開きます。
2. **タスク** ペインで **<<開始>>** ステップをクリックします。
3. **プロパティ** ペインを開きます。
4. **キーワード** を **はい** に設定します。

これで、ビジュアルテストをキーワード駆動テストでキーワードとして使用できるようになります。

スクリプトのテストメソッドをキーワードとして指定

VB .NET スクリプトの既存のテストメソッドをキーワードとして指定して、キーワード駆動テストのメソッドを再利用します。

1. キーワードとして指定するテストメソッドを含む VB .NET スクリプトを開きます。
2. `<Keyword()>` をテストメソッドの直前に追加します。
デフォルトでは、キーワード名はテストメソッドの名前です。
3. 省略可能 : `<Keyword("KeywordName")>` をテストメソッドの直前に追加すると、キーワードに他の名前を設定できます。

これで、テストメソッドをキーワード駆動テストでキーワードとして使用できるようになります。

使用例

テストメソッド `Login` を `Login` という名前で新しいキーワードとして指定するには、テストメソッドの直前に以下を入力します。

```
'VB .NET code
<Keyword("Login")>
```

2つの入力パラメータ `UserName` と `PassWord` を持つテストメソッド `Login` を `Login` という名前で新しいキーワードとして指定するには、以下を入力します。

```
// VB .NET code
<Keyword("Login", Description="Logs in with the given name and
password.")>
Public Sub Login(Argument("UserName") As String, Argument("PassWord")
As String)
... //
End Sub
```



注: ヘルプ トピックを PDF で参照している場合、このサンプルコードは、実際のスクリプトでは許されない場所で改行されてしまっている場合があります。スクリプトでこのサンプルコードを使用する場合は、これらの改行を削除してください。

キーワード間でのデータの受け渡し

キーワード間でパラメータを受け渡し、ビジュアルテストまたは .NET スクリプトとして実装されているキーワードから値を戻すことができます。ビジュアルテストのグローバル変数の値をキーワードに戻すことはできません。これらの変数は、ビジュアルテストの実行エンジンによって管理されており、.NET スクリプトやキーワードからはアクセスできないためです。

1. あるキーワードから他のキーワードにデータを渡したい場合：
 - a) 出力パラメータをデータを渡したいキーワードに追加します。
 - b) 同じパラメータを入力パラメータとしてデータを渡されたいキーワードに追加します。
2. ビジュアルテストまたは .NET スクリプトとして実装されているキーワードからデータを戻したい場合：
 - a) 出力パラメータをデータを戻したいキーワードに追加します。
 - b) データを戻されたいビジュアルテストまたは .NET スクリプトで、戻されたデータを格納するローカル変数を作成します。
 - c) ビジュアルテストまたは .NET スクリプトでローカル変数を使用します。

キーワード駆動テストの編集





注: Silk Test Workbench では、Silk Test Workbench にあるキーワード駆動テストの編集と実行を行うことができ、また、Silk Central に格納されているキーワード駆動テストを実行することができます。Silk Central に格納されているキーワード駆動テストを編集するには、**キーワード駆動テストエディター** でキーワード駆動テストを開き、**編集** をクリックします。

キーワード駆動テストを編集するには：

1. **キーワード駆動テスト エディター** でキーワードを開きます。
 - a) **ウィンドウ > アセットブラウザ** をクリックします。**アセットブラウザ** が開きます。
 - b) **キーワード駆動テスト** をクリックします。
 - c) 編集するキーワード駆動テストをダブルクリックします。
2. 新しいキーワードをキーワード駆動テストに追加するには：
 - a) **新しいキーワード** フィールドをクリックします。
 - b) 新しいキーワードの名前を入力します。
 - c) Enter を押します。
3. 既存のキーワードを編集するには、キーワードの左側にある **キーワードを開く** をクリックします。







注: Silk Central は、Silk Central で作成したすべてのキーワードの所有権を持ちます。このことは、このようなキーワードに対して行う変更は Silk Test Workbench でではなく、Silk Central で保存されることを意味します。



4. キーワードをキーワード駆動テストにコピーするには：
 - a) キーワードを選択します。
 -  **ヒント:** 行番号の列で **Ctrl + クリック** または **Shift + クリック** を使用すると、複数のキーワードを選択できます。
 - b) **Ctrl + C** を押します。
 - c) キーワードを挿入する上の行を選択します。
 - d) **Ctrl + V** を押します。
5. キーワード駆動テストのほかの場所にキーワードを移動するには、キーワードをクリックして他の場所にドラッグするか、または次の手順を実行します。
 - a) キーワードを選択します。
 -  **ヒント:** 行番号の列で **Ctrl + クリック** または **Shift + クリック** を使用すると、複数のキーワードを選択できます。
 - b) **Ctrl + X** を押します。
 - c) キーワードを挿入する上の行を選択します。
 - d) **Ctrl + V** を押します。
6. キーワード駆動テストからキーワードを削除するには、キーワードの左側にある **キーワードの削除** をクリックします。

キーワードは、**キーワード** ウィンドウでまだ利用可能なので、いつでもキーワード駆動テストに再度追加することができます。
7. 変更を保存するには、**新しいバージョンとして保存** をクリックします。

Silk Central でテストのキーワードを管理する

キーワード ページでは、選択したキーワード駆動テストのキーワードを管理できます。次のアクションを実行できます。

タスク	ステップ
テストまたはキーワード シーケンスを Silk Test で開く	Silk Test で開く をクリックして、選択したテストまたはキーワード シーケンスを Silk Test で開きます。
キーワードの追加	<ol style="list-style-type: none"> 1. キーワード リストの一番下にある 新しいキーワード をクリックするか、キーワードを右クリックして、コンテキスト メニューから キーワードを上挿入 を選択します。  注: キーワードの使用状況に基づいて、Silk Test にキーワードを推薦させることができます。推薦させるかどうかを切り替えるには、コンテキスト メニューの レコメンドの有効化 または レコメンドの無効化 を使用します。詳細については、「<i>Silk Test Workbench</i> が推薦するキーワード」を参照してください。 2. 利用可能なキーワードのリストからキーワードを選択するか、新しいキーワードを作成します。 3. 保存 をクリックします。 <p>または、右側にある すべてのキーワード ペインから、既存のキーワードをダブルクリックするか、ドラッグ&ドロップします。</p> <p> ヒント: Ctrl + クリック を使用すると、複数のキーワードを選択できます。ドロップするとき、キーワードは選択した順番に並び替えられます。</p>
キーワードの削除	削除するキーワードの アクション 列で  をクリックします。 保存 をクリックします。
キーワードの順序の変更	キーワードを移動したい位置にドラッグ&ドロップします。 保存 をクリックします。
キーワード シーケンス (他のキーワードから構成されるキーワード) の作成	<ol style="list-style-type: none"> 1. キーワード リストから結合したいキーワードを選択します。行番号の列で Ctrl + クリック または Shift + クリック を使用すると、複数のキーワードを選択できます。 2. 選択範囲を右クリックして、結合 をクリックします。 3. 新しいキーワード シーケンスの 名前 と 説明 を指定します。
キーワード シーケンスからキーワードの抽出	キーワード シーケンスを右クリックし、 キーワードの抽出 を選択します。これによって、元のキーワード シーケンスがそれに含まれるキーワードによって置換されますが、ライブラリからは削除されません。 保存 をクリックします。
テストまたはキーワード シーケンスへのキーワードのコピーと貼り付け	<ol style="list-style-type: none"> 1. キーワード リストからコピーしたいキーワードを選択します。行番号の列で Ctrl + クリック または Shift + クリック を使用すると、複数のキーワードを選択できます。 2. Ctrl + C を押すと選択項目がコピーされます。キーワードを移動する場合には、Ctrl + X を押します。 3. キーワードをコピーするテストまたはキーワード シーケンスを開き、キーワードを挿入する上の行を選択します。 4. Ctrl + V を押します。 <p> ヒント: 選択したキーワードを Excel に貼り付けて編集し、それをコピーしてテストまたはキーワード シーケンスに貼り付けることもできます。</p>
キーワード シーケンスのパラメータの定義	<ol style="list-style-type: none"> 1. キーワード リスト上の パラメータ をクリックします。パラメータ ダイアログ ボックスが表示されます。 2. パラメータの追加 をクリックします。

タスク	ステップ
	<ol style="list-style-type: none"> 3. 新しいパラメータに対して 名前 を指定します。パラメータが出力パラメータ (入力値を必要とするのではなく、値を戻します) の場合、出力 チェックボックスをチェックします。 4. OK をクリックします。 5. 保存 をクリックします。 <p> 注: キーワードまたはキーワード シーケンスは、入力パラメータと出力パラメータを合わせて 20 個まで持つことができます。</p>
ドラフト キーワードの編集	<ol style="list-style-type: none"> 1. 編集するドラフト キーワードのアクション 列で、 をクリックします。 2. キーワードの グループ を選択するか、新しいグループを指定します。 3. 説明 にキーワードの説明を入力します。この情報は、キーワードを実装するエンジニアにとって有益な情報です。 4. OK をクリックします。 5. 省略可能 : パラメータ フィールドをクリックして、キーワードのパラメータを追加します。キーワードが Silk Test で実装されると、これらのパラメータは、生成されたコード スタブに現れます。 6. 保存 をクリックします。
キーワードの検索	<p>キーワード ウィンドウ の検索フィールドを使用して、特定のキーワードを検索します。文字を入力するとリストが更新され、その文字に一致する既存のキーワードが動的に表示されます。検索のヒント :</p> <ul style="list-style-type: none"> • 検索では大文字小文字は区別されません。doAction を入力すると、doaction や DOAction が表示されます。 • 大文字のみを入力すると、いわゆる キヤメルケース検索が実行されます。ECD を入力すると、Enter Car Details、Enter Contact Details、EnterContactDetails が表示されます。 • キーワード名とグループ名が検索対象になります。test を入力すると、test を含むすべてのキーワードと、グループ名に test を含むグループのすべてのキーワードが表示されます。 • ? は、0 または 1 文字と一致します。user?test を入力すると、userTest や usersTest が表示されます。 • * は、0 または n 文字と一致します。my*keyword を入力すると、myKeyword や myNewKeyword や my_other_keyword が表示されます。 • <文字列>. はグループ名のみを検索します。group. を入力すると、グループ名に group を含むグループのすべてのキーワードが表示されます。 • .<文字列> はキーワード名のみを検索します。.keyword を入力すると、keyword を含むすべてのキーワードが表示されます。 • <string>.<string> は特定のグループのキーワードを検索します。group.word を入力すると、myGroup グループの myKeyword が表示されます。 • 引用符を使用して完全一致のみを検索します。'Keyword' を入力すると、Keyword や MyKeyword は表示されますが、keyword は表示されません。

Silk Test Workbench のキーワード レcommend 機能

キーワード駆動テスト エディター でキーワード駆動テストにキーワードを追加するとき、Silk Test Workbench は、そのテストの次のキーワードとして使用する可能性のある既存のキーワードを推薦します。推薦するキーワードはキーワード リストの上位にリストされ、Silk Test Workbench がどの程度そのキーワードを推薦しているかが、棒グラフの塗りつぶした長さによって示されます。

Silk Test Workbench は、次のアルゴリズムに基づいてキーワードを推薦します。


- キーワード駆動テストまたはキーワード シーケンスに最初のキーワードを追加する場合、Silk Test Workbench は他のキーワード駆動テストまたはキーワード シーケンスの最初のキーワードとして使用されているのと同様なキーワードを検索します。最も頻繁に使用されるキーワードがより上位に推薦されます。
- 既に他のキーワードを含んでいるキーワード駆動テストまたはキーワード シーケンスに、さらにキーワードを追加する場合、Silk Test Workbench は次のようにしてキーワードを推薦します。
 - 新しいキーワードを追加するキーワード駆動テストまたはキーワード シーケンスの位置の前にキーワードがある場合、Silk Test Workbench は前のキーワード群と、すべてのほかのキーワード駆動テストとキーワード シーケンスのキーワードの組み合わせとを比較し、最も頻繁に使用されているキーワードの組み合わせに続いて現れるキーワードを推薦します。
 - キーワード駆動テストまたはキーワード シーケンスの位置の前にキーワードがないが、現在の位置の後にキーワードがある場合、Silk Test Workbench は後のキーワード群と、すべてのほかのキーワード駆動テストとキーワード シーケンスのキーワードの組み合わせとを比較し、最も頻繁に使用されているキーワードの組み合わせの前に現れるキーワードを推薦します。
- さらに、Silk Test Workbench は見つかったキーワードの類似性も考慮します。たとえば、2つのキーワードの名前とグループの両方が一致する場合、Silk Test Workbench は名前だけが一致するキーワードよりもこれらをより上位に推薦します。
- Silk Central との接続が確立すると、現在のプロジェクトに対応するキーワード ライブラリに属したキーワード駆動テストに含まれるキーワードも考慮されます。

キーワードでのパラメータの使用

キーワードまたはキーワード シーケンスは、入力パラメータと出力パラメータを合わせて 20 個まで持つことができます。このトピックでは、VB .NET スクリプトとして実装されたキーワードで作業している場合に、Silk Test Workbench を使用してこれらのパラメータを処理する方法について説明します。

キーワード駆動テスト エディター で、キーワードまたはキーワード シーケンスに対して定義されたパラメータを表示し、パラメータの値を編集できます。

キーワード ウィンドウで、キーワードまたはキーワード シーケンス上にマウス カーソルを動かすと、キーワードまたはキーワード シーケンスに割り当てられているパラメータを確認できます。

 **注:** キーワードがビジュアルテストとして実装されている場合、<<開始>> ステップのプロパティ ウィンドウの **入力パラメータ** と **出力パラメータ** セクションで入出力パラメータを指定できます。

単純なキーワードの入力パラメータ

他のテスト メソッドに対するものと同じ方法で、VB .NET スクリプトとして実装された キーワードの入力パラメータを定義して使用することができます。

次のサンプル コードでは、2つの入力パラメータ `userName` と `password` を持つキーワード `SetUserDetails` を定義する方法を説明します。

```
' VB .NET code
<Keyword>
public Sub SetUserDetails(userName As String, password As String)
```

```
...
End Sub
```

単純なキーワードの出力パラメータ

VB .NET スクリプトとして実装されたキーワードの戻り値または 1 つ以上の出力パラメータを定義することができます。また、戻り値と 1 つ以上の出力パラメータを組み合わせて使用することもできます。

次のサンプルコードでは、文字列を返すキーワード `GetText` を定義する方法を説明します。

```
' VB .NET code
<Keyword>
public Function GetText As String
    Return "text"
End Function
```

次のサンプルコードでは、2 つの出力パラメータ `userName` と `password` を持つキーワード `GetUserDetails` を定義する方法を説明します。

```
' VB .NET code
<Keyword>
public Sub GetUserDetails(ByRef userName As String, ByRef password As String)
    userName="name"
    password="password"
End Sub
```

キーワードシーケンスのパラメータ

キーワードシーケンスのパラメータは、**パラメータ** ダイアログ ボックスで定義し、編集できます。このダイアログ ボックスは、**キーワードシーケンス エディター** の **パラメータ** をクリックして開くことができます。

例 : パラメータを取るキーワード

このトピックでは、パラメータを取るキーワードを使用する方法の例を紹介します。キーワードまたはキーワードシーケンスは、入力パラメータと出力パラメータを合わせて 20 個まで持つことができます。

最初の手順として、使用するキーワードを含んだキーワード駆動テストを作成します。これは、キーワード駆動テスト全体を記録するか、新しいキーワード駆動テストを作成してからキーワード駆動テスト エディターでキーワードを追加することによって行います。

この例では、キーワード駆動テストは次のキーワードを含んでいます。

アプリケーションの開始 これは、AUT を開始して基本状態を設定する標準キーワードです。

Login これは、ユーザー名とパスワードで識別される特定のユーザーで AUT にログインするキーワードです。

GetCurrentUser このキーワードは、AUT に現在ログインしているユーザーの名前を返します。

AssertEquals このキーワードは、2 つの値を比較します。

Logout これは、AUT からユーザーをログアウトするキーワードです。

次のステップでは、パラメータをキーワードに追加します。これを行うには、キーワードのテストスクリプトを開き、メソッドにパラメータを追加します。

入力パラメータ `UserName` と `Password` をキーワード `Login` に追加するには、

```
' VB .NET code
<Keyword>
public Sub Login()
```

```
...  
End Sub
```

を、次のように変更します。

```
' VB .NET code  
<Keyword>  
public Sub Login(UserName As String, Password As String)  
...  
End Sub
```

出力パラメータ `UserName` をキーワード `GetCurrentUser` に追加するには、

```
' VB .NET code  
<Keyword>  
public Sub GetCurrentUser()  
...  
End Sub
```











を、次のように変更します。

```
' VB .NET code  
<Keyword>  
public Sub GetCurrentUser(ByRef currentUser As String)  
...  
End Sub
```

キーワード駆動テスト エディター 上のキーワード駆動テストは、次のようになります。

		Keyword	Parameters	
1	 	Start application		
2	 	Login	<i>UserName</i>	<i>Password</i>
3	 	GetCurrentUser	<i>currentUser ←</i>	
4	 	AssertEquals	<i>Expected</i>	<i>Actual</i>
5	 	Logout		

これで、**キーワード駆動テスト エディター** 上で入力パラメータの実際の値を指定できます。キーワード `GetCurrentUser` の出力パラメータ `UserName` の値を取得するには、`${current user}` のように、変数を指定します。その後のキーワードで、変数を指定して格納された値を渡すことができます。


		Keyword	Parameters	
1	 	Start application		
2	 	Login	<i>UserName</i>	<i>Password</i>
3	 	GetCurrentUser	<code>\${current user}</code>	
4	 	AssertEquals	John Smith	<code>\${current user}</code>
5	 	Logout		

キーワードのキーワード シーケンスへの結合

キーワード駆動テスト エディター を使って、複数のキーワード駆動テストで順番に実行したいキーワードを結合してキーワード シーケンスを作成できます。

1. 結合するキーワードが含まれているキーワード駆動テストを開きます。
2. キーワード駆動テスト エディター で、Ctrl キーを押しながら結合したいキーワードをクリックします。
3. 選択範囲を右クリックして、**結合** をクリックします。**キーワードの結合** ダイアログ ボックスが開きます。
4. **名前** フィールドに、新しいキーワード シーケンスの名前を入力します。
5. 省略可能：**説明** フィールドに、新しいキーワード シーケンスの説明を入力します。
6. **結合** をクリックします。


新しいキーワード シーケンスが開き、**キーワード** ウィンドウにも表示されます。キーワード駆動テストでキーワード シーケンスを使用できます。

 **注:** 他のキーワードと同じように、キーワード シーケンス自身を実行することはできませんが、キーワード駆動テストの一部として実行することができます。

変数を指定したキーワード駆動テストの再生

キーワード駆動テスト用の変数の値を設定する前に、キーワード駆動テストを作成する必要があります。











1. 変数を指定して実行するキーワード駆動テストを開きます。
2. キーワード駆動テスト エディター で、**グローバル変数** をクリックします。**グローバル変数** ダイアログ ボックスが開きます。
3. **変数名** 列で、最初の空のフィールドに新しい変数の名前を入力します。
4. 変数の値を **値** 列に入力します。

 **注:** 値フィールドを空のままにすると、誤ったテスト結果が生じる可能性があります。

5. テストの実行に使用するすべての変数を指定するまで、前のステップを繰り返します。
6. **OK** をクリックします。
7. 実行するキーワード駆動テストを開きます。
8. キーワード駆動テスト エディター で、新しい変数を使用するパラメーターを編集します。
次のように記述します。

```
${variable name}
```

たとえば、次のキーワード駆動テストでは、`${current user}` パラメーターはグローバル変数を使用します。

		Keyword	Parameters	
1	 	Start application		
2	 	Login	<i>UserName</i>	<i>Password</i>
3	 	GetCurrentUser	<code>\${current user}</code>	
4	 	AssertEquals	John Smith	<code>\${current user}</code>
5	 	Logout		

キーワード駆動テストが Silk Test Workbench から実行されるときはいつでも、変数が使用されます。

Silk Test Workbench と Silk Central の統合


Silk Test Workbench と Silk Central を統合することによって、技術者と非技術者のユーザー間で共同作業を行えます。

Silk Test Workbench と Silk Central が統合され、Silk Central に存在するライブラリとアクティブな Silk Test Workbench プロジェクトが同じ名前であれば、**キーワード ビュー** (表示 > **キーワード** から開く) には、アクティブな Silk Test Workbench プロジェクトで定義されたキーワードに加えて、Silk Central ライブラリのすべてのキーワードが表示されます。

 **注:** Silk Central の接続情報は、すべての Silk Test Workbench ユーザーに別々に保存されるため、Silk Central のキーワードおよびキーワード シーケンスで作業するすべての Silk Test Workbench ユーザーは、Silk Test Workbench を Silk Central と統合する必要があります。

Silk Test Workbench と Silk Central を統合すると、次のメリットがあります。

- テスト管理と実行を Silk Central で処理できる
- キーワードが Silk Central データベースに格納され (ライブラリのアップロード)、Silk Central のすべてのプロジェクトで利用できる
- 手動テストを Silk Central で直接自動化し、作成したキーワード駆動テストを Silk Central から Silk Test Workbench で実行できる


 **注:** Silk Test Workbench では、Silk Test Workbench にあるキーワード駆動テストの編集と実行を行うことができ、また、Silk Central に格納されているキーワード駆動テストを実行することができます。Silk Central に格納されているキーワード駆動テストを編集するには、**キーワード駆動テスト エディター** でキーワード駆動テストを開き、**編集** をクリックします。

1. メニューから、**ツール > オプション** を選択します。 **オプション** ダイアログ ボックスが開きます。
2. **オプション** ツリーで、**Silk Central** を選択します。
3. **URL** フィールドに、Silk Central サーバーの URL を入力します。

たとえば、Silk Central のサーバー名が *sctm-server* で、Silk Central のポート番号が 13450 の場合は、`http://sctm-server:13450` と入力します。

4. 認証用 Web サービス トークンを指定します。

Web サービス トークンは、Silk Central の **ユーザー設定** ページから生成できます。このページは、Silk Central のメニューに表示されているユーザー名をクリックするとアクセスできます。

 **注:** Silk Central ユーザー名とパスワードを使って認証を行う場合は、**認証** リストから **ユーザー名とパスワード** を選択します。ただし、セキュリティ上の理由から、ユーザー名とパスワードをネットワーク越しに送信するのではなく、Web サービス トークンを使用することを Micro Focus では推奨しています。

5. **検証** をクリックして、Silk Test Workbench が指定したユーザーで Silk Central サーバーにアクセスできるかどうか確認します。
6. **OK** をクリックします。


Silk Central からキーワードを開く

Silk Central から Silk Test Workbench でキーワードを開くには、Silk Central でキーワードの名前の左にあるアイコンをクリックします。これにより、キーワードを実装する資産が Silk Test Workbench で開きます。

Silk Test Workbench のインスタンスが実行されていない場合は、ユーザー名とパスワードを入力するか、Windows 認証を使用してログインする必要があります。Silk Test Workbench は DSN を作成し、**ログイ**

ンダイアログでそれを自動的に選択します。キーワードが Silk Central にアップロードされたときのデータベースについての接続情報が含まれています。

Silk Test Workbench のインスタンスが実行されており、キーワードが Silk Central にアップロードされたときのデータベースと接続されている場合、キーワードはこのインスタンスで開かれます。


 **注:** キーワード ライブラリが Silk Test Workbench 16.0 から Silk Central にアップロードされたものである場合、Silk Test Workbench でキーワードを開く前にライブラリを再アップロードする必要があります。ライブラリを再アップロードしない場合、キーワードを使用する際に予期しない動作が発生する可能性があります。

Silk Test Workbench での Silk Central キーワードの実装

Silk Central キーワードを実装する前に、Silk Central でキーワード駆動テストの一部としてキーワードを定義します。

Silk Test Workbench で Silk Central キーワードを実装するには：

1. Silk Test Workbench のプロジェクトをキーワード駆動テストを含んだ Silk Central のキーワード ライブラリと同じ名前で作成します。
2. Silk Central のキーワード ライブラリにタイプが割り当てられていない場合、**ツール > キーワード ライブラリのアップロード** をクリックして、ライブラリ タイプを設定します。
3. 省略可能：Silk Central の特定のキーワードを Silk Test Workbench で実装するには、ライブラリの **キーワード** タブを Silk Central で開き、キーワードの **アクション** 列から **Silk Test で実装** をクリックします。
4. Silk Test Workbench で、**表示 > キーワード** をクリックします。
5. **キーワード** ウィンドウで、キーワード駆動テストをダブルクリックします。
キーワード ウィンドウで Silk Central による変更を更新するには、**更新** をクリックします。
6. ツールバーで、**操作の記録** をクリックします。
7. 現在のプロジェクトに対してアプリケーション構成が設定されており、Web アプリケーションをテストする場合、**ブラウザの選択** ダイアログ ボックスが開きます。
 - a) ブラウザーを選択します。
 - b) 省略可能：あらかじめ定義されたブラウザ サイズを使用してデスクトップ ブラウザー上の Web アプリケーションをテストする場合は、**ブラウザ サイズ** リストからブラウザ サイズを選択します。
たとえば、Apple Safari 上の Web アプリケーションを Apple iPhone 7 の画面と同じ大きさのブラウザ ウィンドウでテストするには、リストから **Apple iPhone 7** を選択します。
 - c) 省略可能：ブラウザ ウィンドウの **向き** を選択します。
 - d) 省略可能：**ブラウザ サイズの編集** をクリックすると、新しいブラウザ サイズを指定したり、**ブラウザ サイズ** リストに表示するブラウザ サイズを選択することができます。
8. **記録** をクリックします。
記録の詳細については、「[キーワードの記録](#)」を参照してください。
9. 最初の未実装のキーワードの操作を記録します。
- 10 現在のキーワードに対するすべての操作を記録し終わったら、**次のキーワード** をクリックします。
- 11 **記録中** ウィンドウでキーワードを切り替えるには、**前のキーワード** と **次のキーワード** をクリックします。
- 12 **停止** をクリックします。 **記録完了** ダイアログ ボックスが開きます。

 **注:** Silk Central のキーワード駆動テストは、Silk Test Workbench では読み取り専用であるため、キーワードを削除したり、キーワードの順番を変更することはできません。

Silk Central へのキーワード ライブラリのアップロード

Silk Central で作業するためには、有効な Silk Central の場所が設定されている必要があります。詳細については、「[Silk Test Workbench と Silk Central の統合](#)」を参照してください。

Silk Central で自動テストを自動化するために、Silk Test Workbench プロジェクトで実装したキーワードをキーワード ライブラリとして Silk Central にアップロードできます。そして、キーワードを使用して手動テストを自動化できます。

1. メニューで、**ツール > キーワード ライブラリのアップロード** を選択します。現在の Silk Test Workbench データベースにあるすべてのプロジェクトが表示されます。
2. Silk Central にアップロードするキーワードを含むプロジェクトを選択します。



注: プロジェクトと同じ名前のキーワード ライブラリが Silk Central 存在していることを確認します。

3. **OK** をクリックします。
4. 省略可能：キーワード ライブラリに対する変更の説明を指定します。
5. 省略可能：**設定** をクリックして、Silk Central への接続を設定します。
6. 省略可能：接続した Silk Central インスタンスで利用可能なライブラリを確認するには、リンクをクリックします。
7. **アップロード** をクリックします。



注意: Silk Central のキーワード ライブラリが既に他の自動化ツール、つまり他の Silk Test クライアントに割り当てられている場合、キーワード ライブラリのタイプを変更するかどうか確認されます。タイプの変更を行う場合にのみ、ライブラリはアップロードされます。

Silk Test Workbench は、プロジェクトで実装されたすべてのキーワードからキーワード ライブラリを作成します。その後、Silk Test Workbench は library.zip という名前で、キーワード ライブラリをプロジェクトの出力フォルダに保存します。ライブラリの整合性が検証され、Silk Central の既存のテストが影響を受ける変更が **キーワード ライブラリを Silk Central にアップロード** ダイアログ ボックスに一覧されます。最後に、Silk Test Workbench はライブラリを Silk Central にアップロードします。これで、Silk Central でキーワードを使用できるようになります。キーワード ライブラリに含まれるキーワードを使用する Silk Central のキーワード駆動テストは、現在のキーワードの実装を自動的に使用します。

現在の Silk Test Workbench データベースが Access データベースの場合、アップロードしたキーワードは、Access データベースにアクセスしたマシン上でのみ実行することができます。Silk Central にアップロードされたキーワード ライブラリには、キーワードの実装は含まれず、キーワードがこの Silk Test Workbench データベースで利用可能であるという情報のみ含まれます。

コマンド ラインから Silk Central へのキーワード ライブラリの更新

Java ベースのコマンド ラインから外部キーワード ライブラリを Silk Central にアップロードすることで、Silk Central とキーワード駆動テストを Jenkins などの継続的インテグレーション ビルド システムに統合することができます。

Java ベースのコマンド ラインからキーワード ライブラリを Silk Central にアップロードするには：

1. Silk Central で、**ヘルプ > ツール** を選択し、**Java キーワード ライブラリ ツール** をダウンロードします。
2. ダウンロードした jar ファイルに含まれるコマンド ライン ツールを次の引数で呼び出します。
 - java

- -jar com.borland.silk.keyworddriven.jar
- -upload
- 更新または存在しない場合は作成される Silk Central のライブラリの ライブラリ名。
- アップロードするライブラリ パッケージ (zip アーカイブ) の パッケージ名。
- Silk Central フロントエンド サーバーのホスト名:ポート。
- Silk Central ユーザーの Web サービス トークン。認証に必要です。Web サービス トークンは、Silk Central の **ユーザー設定** ページから生成できます。このページは、Silk Central のメニューに表示されているユーザー名をクリックするとアクセスできます。



注: セキュリティ上の理由から、ユーザー名とパスワードをネットワーク越しに送信するのではなく、Web サービス トークンを使用することを Micro Focus では推奨しています。

- Silk Central ユーザーの ユーザー名。Web サービス トークンを認証に使用する場合は必要ありません。
- Silk Central ユーザーの パスワード。Web サービス トークンを認証に使用する場合は必要ありません。
- ライブラリに適用された変更を説明する 更新情報 (引用符で囲む)。
- テストまたはキーワード シーケンスで使用されているキーワードの削除を許可するための [-allowUsedKeywordDeletion] フラグ (省略可能)。デフォルトでは、使用中のキーワードを削除しようとするとエラーが発生します。

Java 9 以降を使ってライブラリを Silk Central にアップロードするコマンド ラインは次のようになります。

```
java --add-modules=java.activation,java.xml.ws -jar com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scLogin
scPassword "Build xy: Implemented missing keywords"
```

使用例

認証に Web サービス トークンを使い、Java 8 以前を使ってライブラリを Silk Central にアップロードするコマンド ラインは次のようになります。

```
java -jar com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scToken "Build xy:
Implemented missing keywords"
```

認証にユーザー名とパスワードを使い、Java 8 以前を使って上記と同じライブラリをアップロードするコマンド ラインは次のようになります。

```
java -jar com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scLogin
scPassword "Build xy: Implemented missing keywords"
```

Java 9 以降を使った場合は、それぞれ次のようになります。

```
java --add-modules=java.activation,java.xml.ws -jar
com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scToken "Build xy:
Implemented missing keywords"
```

```
java --add-modules=java.activation,java.xml.ws -jar
com.borland.silk.keyworddriven.jar -upload
"My library" "./output/library.zip" silkcentral:19120 scLogin
scPassword "Build xy: Implemented missing keywords"
```



注: Java 9 以降を使ってキーワード駆動ライブラリをアップロードする場合は、実行サーバー上で JAVA_HOME が定義され、対応するバージョンの JDK を指していることを確認してください。

キーワードの検索

キーワード ウィンドウの検索フィールドを使用して、特定のキーワードを検索します。文字を入力するとリストが更新され、その文字に一致する既存のキーワードが動的に表示されます。検索のヒント：

- 検索では大文字小文字は区別されません。doAction を入力すると、doaction や DOAction が表示されます。
- 大文字のみを入力すると、いわゆる キヤメルケース検索が実行されます。ECD を入力すると、Enter Car Details、Enter Contact Details、EnterContactDetails が表示されます。
- キーワード名とグループ名が検索対象になります。test を入力すると、test を含むすべてのキーワードと、グループ名に test を含むグループのすべてのキーワードが表示されます。
- ? は、0 または 1 文字と一致します。user?test を入力すると、userTest や usersTest が表示されます。
- * は、0 または n 文字と一致します。my*keyword を入力すると、myKeyword や myNewKeyword や my_other_keyword が表示されます。
- <文字列>. はグループ名のみを検索します。group. を入力すると、グループ名に group を含むグループのすべてのキーワードが表示されます。
- .<文字列> はキーワード名のみを検索します。.keyword を入力すると、keyword を含むすべてのキーワードが表示されます。
- <string>.<string> は特定のグループのキーワードを検索します。group.word を入力すると、myGroup グループの myKeyword が表示されます。
- 引用符を使用して完全一致のみを検索します。'Keyword' を入力すると、Keyword や MyKeyword は表示されますが、keyword は表示されません。

キーワードのフィルタリング

現在のプロジェクトの特定のキーワードを検索するために、**キーワード** ウィンドウに表示されているキーワードをフィルタすることができます。Silk Central との統合が設定されている場合、結果には Silk Central に関連付けられたキーワードも含まれます。

1. メニューで **表示 > キーワード** をクリックします。**キーワード** ウィンドウが開きます。
2. **キーワード** ウィンドウで、検索フィールドに検索するキーワードの名前を入力します。**キーワード** ウィンドウに、現在のプロジェクト内の指定した名前を持つすべてのキーワードが表示されます。
3. 省略可能：キーワードを編集するには、**キーワード** ウィンドウでキーワード上にマウスカーソルを移動して、**実装へ移動** をクリックします。

キーワードのグループ化

キーワードをライブラリで構造化するために、グループ化することができます。

このトピックでは、キーワードを特定のグループに追加する方法について説明します。このグループ名は Silk Central でも使用され、キーワードはグループ名に従って分類されます。

キーワードを特定のグループに追加するには：

1. キーワードの実装を開きます。
 - a) キーワードを実装しているプロジェクトをアクティブ化します。
 - b) **キーワード** ウィンドウを開きます。
 - c) **キーワード** ウィンドウで、キーワードを選択します。
 - d) **実装へ移動** をクリックします。

2. キーワードが VB .NET スクリプトで実装されている場合は、キーワード グループをスクリプトの最初に追加します。

たとえば、キーワードを Calculator グループ追加するには、次のように入力します。

```
<KeywordGroup("Calculator")>
```

3. キーワードがビジュアル テストで実装されている場合は、キーワード グループをビジュアル テストのプロパティに追加します。

a) ビジュアル テストの <<開始>> ステップを選択します。

b) プロパティ ペインを開きます。

c) キーワード グループ プロパティに、グループの名前を入力します。

キーワード ウィンドウで表示されるキーワード名にグループが含まれるようになります。たとえば、Addition キーワードが Calculator グループに属している場合は、Calculator.Addition として表示されます。

キーワード駆動テストのトラブルシューティング

Silk Central からキーワード駆動テストの再生時に「アプリケーション構成が存在しません」エラーが発生する理由

このエラーが発生した場合、キーワード駆動テストの最初のキーワードとして アプリケーションの開始 キーワードが含まれていません。Silk Test Workbench では、キーワード駆動テストにプロジェクトのアプリケーション構成を適用するには、アプリケーションの開始 キーワードが必要です。新しいキーワード駆動テストを記録する場合には、Silk Test Workbench は、アプリケーションの開始キーワードを最初のキーワードとして自動的にキーワード駆動テストに追加します。

Silk Test Workbench 19.5 以降では、並列テストはデフォルトで有効化されています。Silk Test Workbench 19.5 より前のバージョンから Silk Test Workbench 19.5 以降のバージョンに Silk Test Workbench をアップグレードすると、以前は実行できていたキーワード駆動テストが実行できなくなる可能性があります（各テストごとにテスト対象アプリケーションを指定する必要があるため）。

この問題を回避するには、Silk Test Workbench を実行するマシン上で環境変数 `SILKTEST_ENABLE_PARALLEL_TESTING` を `false` に設定して、並列テストを無効化してください。

ブラウザやモバイル アプリがテストとテストの間で閉じないようにする方法

Silk Test Workbench 19.5 以降を使って Silk Central から複数のキーワード駆動テストを再生すると、デフォルトで並列テストが有効化されます。Silk Test Workbench 19.5 より前のバージョンの Silk Test Workbench から、より新しいバージョンの Silk Test Workbench にアップグレードすると、Silk Test Workbench はデフォルトでキーワード駆動テストの完了時にブラウザやモバイル アプリを閉じます。

この問題は、Internet Explorer 上でブラウザ アプリケーションをテストする場合には発生しません。この問題を回避するには、Silk Test Workbench を実行するマシン上で環境変数 `SILKTEST_ENABLE_PARALLEL_TESTING` を `false` に設定して、並列テストを無効化してください。

コマンドライン

Silk Test Workbench には、テストを開いて実行するためのコマンドラインプログラムが 2 つ含まれています。

SilkTest.Exe コマンドライン

SilkTest.exe コマンドラインプログラムを使用すると、ログインダイアログボックスをスキップして Silk Test Workbench を開くことができます。これは ¥Silk¥Silk Test¥ng¥gui¥ にあります。

パラメータ

パラメータの長い名前	パラメータの短い名前	説明
userid または user	-u	ログインに使用するユーザー名。
pass または password	-p	ログインユーザーのパスワード。
dsn	-d	使用する DSN。

例

```
SilkTest.exe -u admin -p mypassword -d mydsn
```

STW.EXE コマンドライン

Silk Test Workbench には、バッチスクリプト実行の 1 つ以上の .NET スクリプトまたはビジュアルテストを実行するために使用できるコマンドラインプログラムが含まれています。コマンドラインはエラー情報を返し、オプションで結果の出力ファイルを作成します。

コマンドラインプログラムの名前は stw.exe であり、次の場所にあります : ¥Silk¥Silk Test¥ng¥gui¥.

コマンドラインの構文は次のとおりです :

```
STW -d DSN_Name -u User_Name -p User_Password -s Script_Name
```

このセクションのトピックでは、必須パラメータとオプションパラメータを指定してプログラムを実行する方法について説明します。

STW.EXE のパラメータ

コマンドラインプログラムには、必須パラメータとオプションパラメータの両方があります。パラメータは、長い形式と短い形式のどちらかの名前を使用して指定できます。一部のオプションパラメータでは、環境の変更、スクリプトに変数を渡したり、出力ファイルの作成などを指定できます。






注: パラメータ名、スクリプト、スクリプトパラメータでは、大文字と小文字が区別されません。ただし、スクリプトパラメータの値は、指定どおりにスクリプトに渡されます。




注: パラメータ名の前には、文字 - の代わりに文字 / を使用できます。

パラメータの長い名前	パラメータの短い名前	必須かどうか	説明
dsn	-d	はい	データベースへの接続に使用する DSN の名前。このパラメータは必須ですが、空にすると <i>SilkTest</i> が使用されます。
username	-u	はい	接続に使用されるユーザー名。空にすると <i>Admin</i> が使用されます。
password	-p	いいえ	接続に使用されるパスワード名。パスワードが <i>Nothing</i> の場合はオプション (お勧めできません)。
project	-r	いいえ	再生するスクリプトを含むプロジェクト。デフォルトでは、プロジェクトのすべてのスクリプトが実行されます。固有のスクリプトを指定するには、 <code>-script</code> パラメータを使用します。スクリプトを指定せず、スクリプトで入力パラメータが必要である場合は、デフォルト値が使用されます。 .NET スクリプトが先に実行され、ビジュアルテストが次に実行されます。 プロジェクトが存在しない場合は、エラーメッセージが表示されます。  重要: プロジェクトパラメータが指定されていない場合、 <i>Common</i> プロジェクトがデフォルトで使用されます。たとえば、「 <code>stw -d silktest -u admin -p admin -s Script1 -r OtherProject -s Script2</code> 」のように指定した場合は、 <i>Script1</i> は <i>Common</i> プロジェクトで実行され、 <i>Script2</i> は <i>OtherProject</i> という名前のプロジェクトで実行されます。
environment	-e	いいえ	スクリプトに使用する実行環境の名前。このパラメータを含めない場合は、デフォルトの再生環境が使用されます。 再生環境が存在しない場合、スクリプトは実行されません。  重要: 使用する場合は、このパラメータの次に <code>project</code> と <code>script</code> パラメータを指定する必要があります。
script	-s	はい	再生する .NET スクリプト、ビジュアルテスト、またはキーワード駆動テストの名前。 .NET スクリプト、ビジュアルテスト、またはキーワード駆動テストが存在しない場合、エラーメッセージが表示されます。
variable	-m	いいえ	このパラメータを使用すると、.NET スクリプト、ビジュアルテスト、キーワード駆動テストのパラメータを指定できます。たとえば、このパラメータを使用して、 <i>Silk Central</i> からキーワード駆動テストにグローバルプロパティを渡すことができます。 指定すると、資産で定義されたグローバルプロパティは上書きされます。このパラメータを使用する場合は、 <code>script</code> パラメータの次に指定する必要があります。 名前と値のペアは、二重引用符で囲む必要があります。変数に割り当てる値を表すには、文字 <code>=</code> を使用します。ブール値パラメータは、 <i>True</i> または <i>False</i> として指定する必要があります。

パラメータの長い名前	パラメータの短い名前	必須かどうか	説明
file	-f	いいえ	<p>パラメータは任意の順序で指定できますが、名前と数はスクリプトのパラメータと正確に一致する必要があります。渡す名前が多すぎたり、少なかったり、無効だったりすると、エラーメッセージが表示されてスクリプトは実行されません。</p> <p>このオプションでは、コマンドラインパラメータをテキストファイルで指定し、コマンドラインプログラムに渡すことができます。再入力したくない長い引数がある場合にこれを使用できます。</p> <p>使用法の詳細については、「入力ファイル」を参照してください。</p>
outputfile	-o	いいえ	<p>実行されたスクリプトに関する情報を含むファイルの名前を指定するには、このパラメータを使用します。実行されて、正常に完了したか再生エラーを生成したスクリプトのみが出力ファイルに表示されます。</p> <p>パスや名前にスペースが含まれる場合は、出力ファイルの名前を引用符で囲む必要があります。</p> <p>使用法の詳細については、「出力ファイル」を参照してください。</p>
browser	-b	いいえ	<p>ブラウザー アプリケーション構成を持つスクリプトが実行されるブラウザーを指定するには、このパラメータを使用します。パラメータには次の値を指定できます (文字小文字は区別されません)。</p> <ul style="list-style-type: none"> • InternetExplorer • Firefox • GoogleChrome • Edge • AndroidBrowser • Safari • None <p>AndroidBrowser または Safari を使用する場合には、mobiledevice パラメータを指定します。</p> <p> 重要: 使用する場合は、このパラメータの次に project と script パラメータを指定する必要があります。</p>
commandlineargs	-cla	いいえ	<p>スクリプトの基本状態に渡すコマンドライン引数を指定する場合には、このパラメータを使用します。コマンドライン引数を使用する任意の基本状態に対して、このパラメータを使用できます。エージェントがコマンドライン引数を使用する場合、この設定が渡されますが、それ以外の場合は、無視されます。</p> <p>コマンドライン引数の指定には、<i>silktest.commandLineArgs</i> 環境変数を使用することもできます。STW.exe のパラメータと環境変数の両方でコマンドライン引数が指定された場合は、STW.exe のパラメータが使用されます。</p> <p> 重要: 使用する場合は、このパラメータの次に script パラメータを指定する必要があります。</p>
viewportwidth	-vw	いいえ	<p>ブラウザー ビューポートの幅をピクセルで指定する場合には、このパラメータを使用します。viewportwidth パラメータ</p>

パラメータの長い名前	パラメータの短い名前	必須かどうか	説明
viewportheight	-vh	いいえ	と viewportheight パラメータは同時に指定する必要があります。いずれか一方のパラメータだけを指定すると、エラーになります。viewportname パラメータを、このパラメータと同時に指定することはできません。
viewportname	-vn	いいえ	ブラウザ ビューポートの高さをピクセルで指定する場合には、このパラメータを使用します。viewportheight パラメータと viewportwidth パラメータは同時に指定する必要があります。いずれか一方のパラメータだけを指定すると、エラーになります。viewportname パラメータを、このパラメータと同時に指定することはできません。
orientation	-on	いいえ	UI であらかじめ定義したカスタム ブラウザー サイズのリストの名前を使用して、ビューポートの幅と高さを指定する場合には、このパラメータを使用します。このリストは、 ブラウザーの選択 ダイアログ、 アプリケーションの選択 ダイアログの Web タブ、 ブラウザー アプリケーション構成の編集 ダイアログで、ブラウザーを選択するときに表示されます。viewportwidth パラメータと viewportheight パラメータを、このパラメータと同時に指定することはできません。
mobiledevice	-md	いいえ	orientation パラメータは、viewportname が指定されたときだけ有効になります。その値として、 <i>portrait</i> または <i>landscape</i> を指定できます。UI の 向き の設定に該当します。
append	-a	いいえ	スクリプトを実行するモバイル デバイスの名前。この名前は、 ブラウザーの選択 ダイアログ ボックスにも表示されます。 この値をリセットするには、二重引用符で囲んだ空の文字列 ("") を渡します。  重要: 使用する場合は、このパラメータの次に project と script パラメータを指定する必要があります。
increment	-i	いいえ	結果を追加するには、このパラメータを使用します。 オプション ダイアログ ボックスに表示されるものと同じ設定です。
verbose	-v	いいえ	結果を増分するには、このパラメータを使用します。 オプション ダイアログ ボックスに表示されるものと同じ設定です。
help または ?	-h	いいえ	このパラメータでは、スクリプトの実行中にユーザーに出力が表示されます。コマンド ラインの任意の場所に表示されることがあります。 DSN、ユーザー名、パスワードは常に表示されます。出力ファイルを使用する場合は、それが表示されます。スクリプト名が表示される場所では、プロジェクト名が表示され、名前が括弧で囲まれて後に続きます。スクリプトの名前、プロジェクト、スクリプト ステータス、合計検証数、合格した検証数、失敗した検証数を含む行は、実行されたスクリプトごとに表示されます。
resultdir	-rd	いいえ	コマンドラインプログラムの使用可能パラメータを表示するには、「STW -?」と入力します。
			実行したスクリプトのすべての結果ファイルを保存するディレクトリを指定するには、このパラメータを使用します。 次のファイルがこのディレクトリに保存されます。 <ul style="list-style-type: none">XML 結果ファイル

パラメータの長い名前	パラメータの短い名前	必須かどうか	説明
resultformat	-rf	いいえ	<ul style="list-style-type: none"> XML 結果ファイルに対応する XSLT ファイル。resultformat パラメータの値に JUnit が指定された場合は生成されません。 エラー画面 (実行中にエラーが発生した場合) <p>ファイル名は、プロジェクト名、結果名、バージョン、データの種類の組み合わせになります。たとえば、ProjectA.resultcomment.ver12.Info.xml や ProjectA.resultcomment.ver12.Result.xml になります。</p> <p>resultdir パラメータと一緒にこのパラメータを使用して、結果から生成される XML ファイルの形式を指定します。</p> <p>以下の値を指定できます。</p> <p>Workbench ブラウザーに結果を表示するために使用する結果のエンティティと XSLT ファイルを含む XML ファイルを作成します。</p> <p>JUnit JUnit 用の形式で結果を保存した XML ファイルを作成します。</p> <p>デフォルト値は、Workbench です。</p> <p> 注: resultformat パラメータを resultdir パラメータを指定せずに使用した場合は、resultformat パラメータの値は無視されます。</p>
connectionstring	-c	いいえ	<p>リモート ロケーションに接続されているブラウザやモバイル デバイスをテストするには、このパラメータを使用します。connectionstring パラメータを使用してテストを実行するとき、モバイル デバイスとパラメータの間で connectionstring は出力ファイルに書き出されます。</p>

例 1

この例では、Internet Explorer でブラウザ スクリプトを実行するコマンドを示します。

```
STW -dsn silktest -browser InternetExplorer -script MyScript
```

例 2

この例では、Internet Explorer と Google Chrome の両方でブラウザ スクリプトを実行するコマンドを示します。

```
STW -dsn silktest -browser InternetExplorer -script MyScript -browser GoogleChrome -script MyScript
```


例 3

この例では、まず、Android ブラウザーでスクリプトを実行し、それからスクリプトのアプリケーション構成で設定されたブラウザで実行するコマンドを示します。

```
STW -dsn silktest -browser AndroidBrowser -mobiledevice Nexus_4 -script MyScript -browser None -mobiledevice "" -script MyScript
```

例 4

この例では、モバイル デバイス上の Chrome for Android で Common プロジェクトのすべてのスクリプトを実行するコマンドを示します。

```
STW -dsn silktest -browser GoogleChrome -mobiledevice Nexus_4 -project Common
```

例 5

この例では、Mac に接続されたモバイル デバイス上の Chrome for Android で Common プロジェクトのすべてのスクリプトを実行するコマンドを示します。

```
STW -dsn silktest -connectionstring "platformName=Android;deviceName=Moto G;host=http://MyMac -GoogleChrome" -project Common
```

例 6

この例では、ブラウザ ビューポート サイズ 800x600 の Internet Explorer でブラウザ スクリプトを実行するコマンドを示します。

```
STW -dsn silktest -browser InternetExplorer -viewportwidth 800 -vieportheight 600 -script MyScript
```

例 7

この例では、ブラウザ ビューポート サイズ SVGA (800, 600) の Internet Explorer でブラウザ スクリプトを実行するコマンドを示します。

```
STW -dsn silktest -browser InternetExplorer -viewportname SVGA -script MyScript
```

例 8

この例では、カスタム プロファイルを指定した Mozilla Firefox でブラウザ スクリプトを実行するコマンドを示します。

```
STW -dsn silktest -username admin -password admin -browser Firefox -commandLineArgs "-p MyProfile" -project Common -script BrowserTest
```

入カファイル

-file オプションを使用すると、テキスト ファイルでコマンド ライン パラメーターを指定して、コマンド ライン プログラムに送信することができます。

ファイルはコマンド ラインの任意の場所で指定できますが、指定した順番に処理され、その前後のコマンド ライン オプションに影響することがあります。また、複数の -file コマンドを指定（後にファイル名を

指定) することも、ファイル内で `-f` パラメーターを指定することもできます。パラメーターは複数行で指定できます。

行をコメントアウトするには、`#` 文字を使用します。

ファイル内に別の `-file` パラメーターが含まれている場合は、再帰の使用について確認されます。つまり、ファイル内の `-file` パラメーターが、同じ処理セット内の任意の場所で参照されているのと同じファイルを示すことがあってはなりません。

ファイルの有効な形式は、UTF-8、UTF-16、または Unicode です。

例 1 : FileA.txt

```
# Execute my script
-script MyScript
# Change the environment and run again
-environment MyEnvironment
-script MyScript
```

例 2 : FileB.txt

```
# Include contents of FileA.txt, twice
-file FileA.txt

# Reset the playback environment
-environment

# Include it again
-file FileA.txt
```

出力ファイル

`outputfile` パラメータを使用すると、コマンド ラインは指定ファイルを作成して内容を上書きしようとします。ファイルにアクセスできない場合はエラー メッセージが表示されますが、実行は進みます。これが発生する可能性がある理由は次のとおりです。

- パスが存在しない。
- 別のプロセスによってロックされているか使用されている。
- そのフォルダのアクセス権がユーザーにない。

出力ファイル形式

作成されるファイルはテキスト ファイルであり、各行は実行されたスクリプトを表します。各行には多くのフィールドが含まれ、フィールドはタブ文字 (0x9) で区切られます。これは TXT 形式であり、Microsoft Office Excel で使用できます。

出力ファイルのフィールド

フィールド	説明
スクリプト名	.NET またはビジュアル テスト スクリプトのどちらかになります。
プロジェクト	スクリプトが存在するプロジェクト。コマンド ラインにプロジェクトを指定しない場合は、ここに「Common」と表示されます。
実行環境	スクリプトの実行に使用された実行環境の名前。デフォルトの実行環境が使用された場合、このフィールドは空です。
合計検証数	スクリプトによって実行された検証の総数。
合格した検証数	合格した検証の数。

フィールド	説明
失敗した検証数	失敗した検証の数。
スクリプト ステータス	スクリプトのステータス。
再生エラー	再生エラー メッセージが含まれます。再生エラーが検出されない場合は空です。
ブラウザー	スクリプトの実行に提供されたブラウザーの名前。
接続文字列	モバイル デバイスまたはリモート ロケーションへの接続文字列。
スクリプト パラメータ	スクリプトがパラメータを取る場合は、パラメータがフィールドのペアでここにリストされます。最初のフィールドがパラメータ名、次のフィールドが値です。

エラー

コマンド ライン プログラムは ERRORLEVEL をオペレーティングシステムに返し、呼び出し側プログラムはこれを分析して、実行時に発生したことを判断できます。次の表ではエラーについて説明します。

コード	説明	スクリプトのステータスとして出力ファイルで返されるかどうか
0	エラーはありません。すべてのスクリプトは正常に実行されました。	はい
1	致命的なエラー。コマンド ライン プログラムをまったく実行できません。言語リソース ファイルがないことが理由の 1 つと考えられます。	いいえ
2	Silk Test COM オブジェクトを作成できませんでした。COM オブジェクトが登録されたことを確認してください。Silk Test.exe を実行すると COM オブジェクトが登録されることに注意してください。	いいえ
3	DSN、ユーザー名、パスワードのいずれかが正しくありません。詳細については、「 Silk Central から STW.EXE を実行するための前提条件 」を参照してください。	いいえ
4	スクリプト名が指定されていません。	いいえ
5	指定したプロジェクトが見つかりませんでした。	いいえ
6	指定したスクリプトが見つかりませんでした。	いいえ
7	指定した実行環境が見つかりませんでした。	いいえ
8	スクリプトのパラメータが無効です。指定したスクリプトのパラメータが少なすぎるか多すぎるか、パラメータが見つかりませんでした。	いいえ
9	スクリプトの実行が完了しませんでした。最後に達する前に停止した可能性があります。再生エラーという結果にはなっていません。	はい
10	スクリプトで再生エラーが発生しました。	はい
11	スクリプトが実行できなかったか、スクリプトの検証がエラーになりました。	はい
12	複数の問題が発生しました。たとえば、スクリプトがエラーになり、スクリプト再生エラーが検出されました。	いいえ
13	-file オプションのためにコマンド ラインを処理できなかった場合、これが返されます。たとえば、ファイルが存在しないか、ファイルの再帰がある場合です。	いいえ
14	コマンド ライン ウィンドウの停止またはシャットダウンを行う方法はさまざまです。Ctrl+C、Ctrl+Break、Windows からのログアウト、Windows のシャットダウンのいずれかにより、コマンド ラインは閉じるか停止します。このような操作は捕えられて、このエラー コー	いいえ

コード	説明	スクリプトのステータスとして出力ファイルで返されるかどうか
	ドが返されます。ウィンドウを閉じると、スクリプトの停止、製品のログアウト、SilkTest.EXE のアンロードも行われます。	
15	viewportwidth または viewportheight のいずれか一方だけが指定されました。	いいえ
16	viewportname が viewportwidth と viewportheight と同時に指定されました。	いいえ
17	resultformat パラメータの値が無効です。	いいえ

バッチ ファイルを使用して ERRORLEVEL を確認する

バッチ ファイルで ERRORLEVEL を確認する方法は多数あります。次の例で 1 つの方法について説明します。

```
@ECHO OFF
SETLOCAL

REM -----
REM Define the error codes that can be returned by STW.EXE
REM -----
SET eNoError=0
SET eFatalError=2
SET eUnableToCreateComObject=2
SET eLoginFailed=3
SET eNoScriptSpecified=4
SET eProjectNotFound=5
SET eScriptNotFound=6
SET eRunEnvNotFound=7
SET eInvalidScriptParameters=8
SET eScriptDidNotComplete=9
SET eScriptPlaybackError=10
SET eScriptFailed=11
SET eMultipleProblems=12
SET eCommandCouldNotBeProcessed=13
SET ePrograTerminatedByUser=14
SET eInvalidViewportSize=15
SET eInvalidViewportParameters=16

REM -----
REM Run STW.EXE
REM -----
@ECHO ON
STW.EXE -username Admin -dsn STW-Scratch -script ScriptNotFound
@ECHO.
@ECHO OFF

REM -----
REM Anything greater than 12 is unknown, can be caused if the command
REM processor cannot find STW.EXE.
REM -----
IF %ERRORLEVEL% GTR %eMultipleProblems% (
    ECHO STW.EXE returned an unknown return code %ERRORLEVEL%
    GOTO END
)

REM -----
```

```
REM Check the specific error codes here.
REM -----
IF %ERRORLEVEL% EQU %eMultipleProblems% (
    ECHO eMultipleProblems
    GOTO END
)

IF %ERRORLEVEL% EQU %eScriptFailed% (
    ECHO eScriptFailed
    GOTO END
)

IF %ERRORLEVEL% EQU %eScriptPlaybackError% (
    ECHO eScriptPlaybackError
    GOTO END
)

IF %ERRORLEVEL% EQU %eScriptDidNotComplete% (
    ECHO eScriptDidNotComplete
    GOTO END
)

IF %ERRORLEVEL% EQU %eInvalidScriptParameters% (
    ECHO eInvalidScriptParameters
    GOTO END
)

IF %ERRORLEVEL% EQU %eRunEnvNotFound% (
    ECHO eRunEnvNotFound
    GOTO END
)

IF %ERRORLEVEL% EQU %eScriptNotFound% (
    ECHO eScriptNotFound
    GOTO END
)

IF %ERRORLEVEL% EQU %eProjectNotFound% (
    ECHO eProjectNotFound
    GOTO END
)

IF %ERRORLEVEL% EQU %eNoScriptSpecified% (
    ECHO eNoScriptSpecified
    GOTO END
)

IF %ERRORLEVEL% EQU %eLoginFailed% (
    ECHO eLoginFailed
    GOTO END
)

IF %ERRORLEVEL% EQU %eUnableToCreateComObject% (
    ECHO eUnableToCreateComObject
    GOTO END
)

IF %ERRORLEVEL% EQU %eFatalError% (
    ECHO eFatalError
```

```

    GOTO END
)
IF %ERRORLEVEL% EQU %eNoError% (
    ECHO eNoError
    GOTO END
)
IF %ERRORLEVEL% EQU %eCommandCouldNotBeProcessed% (
    ECHO eCommandCouldNotBeProcessed
    GOTO END
)
IF %ERRORLEVEL% EQU %ePrograTerminatedByUser% (
    ECHO ePrograTerminatedByUser
    GOTO END
)
IF %ERRORLEVEL% EQU %eInvalidViewportSize% (
    ECHO eInvalidViewportSize
    GOTO END
)
IF %ERRORLEVEL% EQU %eInvalidViewportParameters% (
    ECHO eInvalidViewportParameters
    GOTO END
)
:END
ENDLOCAL

```

Silk Central から STW.EXE を実行するための前提条件

Silk Central 実行サーバーから STW.EXE を実行し、Silk Test Workbench データベースに正しく接続するには、次の前提条件を満たす必要があります。

1. Silk Central 実行サーバーがサービスではなく、プロセスとして実行されている。
2. Silk Test Workbench データベースへの接続にデータベース認証が使用されている。Silk Test Workbench では、Silk Test Workbench SQL Server データベースとの接続に Windows 認証とデータベース認証を使用できます。
3. データベース サーバー インスタンスが静的ポートを使用するように構成されている。

これらの前提条件が満たされていない場合、Silk Test Workbench 実行サーバーから STW.EXE を実行しようとするときに Silk Central は次のエラーで終了する場合があります。
エラー 3 : DSN、ユーザー名、パスワードのいずれかが正しくありません。

STW.EXE の使用例

コマンドラインでは、STW.EXE の 1 回の実行で任意の数のスクリプトを実行することがサポートされます。コマンドラインでは、入力ファイルと出力ファイル、およびスクリプトごとにプロジェクトと実行環境を変えることもサポートされます。



注: このほとんどの例には、読みやすくするため、-d、-u、-p の各パラメータが含まれていません。

一般的なヘルプ

```
STW -help
```

単一プロジェクト、すべてのスクリプト

```
STW -project ProjectName
```

単一プロジェクト、単一スクリプト

```
STW -project ProjectName -script ScriptName
```

単一スクリプト、Common プロジェクト

```
STW -s scriptname
```

コマンドラインでは、SilkTest を DSN として、Admin をパスワードがないユーザー名として使用します。

名前にスペースが含まれるスクリプト

```
STW -script "Script with spaces"
```

Common プロジェクトの特定スクリプト

```
STW -script Script1 -script Script2
```

Common プロジェクトのすべてのスクリプト

```
STW
```

さまざまなプロジェクトの特定スクリプト

この例では 2 つの別々のスクリプトを実行します。別々のプロジェクトでは同じスクリプト名を使用できるので、この 2 つは 2 つの固有なスクリプトを表します。

```
STW -project Project1 -script Script1 -project Project2 -script Script1
```

Common プロジェクトとユーザー プロジェクトの特定スクリプト



注: プロジェクト名を指定すると、それに続くすべてのスクリプトはそのプロジェクトに関連します。

```
STW -project Project1 -script Script1 -script Script2 -project Common -script Script1 -script Script2
```

入力ファイルの使用

```
STW -file FileA.txt
```

FileA.txt 内に以下が含まれる場合

```
# Execute my script
-script MyScript
# Change the environment and run again
-environment MyEnvironment
-script MyScript
```

環境

```
STW -environment PlaybackEnvironment -s ScriptName
```

複数の実行環境

この例では、ユーザーが定義した実行環境とデフォルトの実行環境で同一スクリプトを実行します。

```
STW -environment MyRunEnv -script Script1 -environment -script Script1
```

-environment の 2 回目の使用では名前を指定しません。環境はデフォルトにリセットされます。

変数

パラメータを必要とする .NET スクリプト、ビジュアルテスト、またはキーワード駆動テストをコマンドラインから実行する場合、*variable* パラメータを使用して資産のパラメータを指定できます。指定すると、資産で定義されたグローバルプロパティは上書きされます。

```
STW -script ScriptWithParameters -variable "var1=1234" "var2=Hello" "var3=True"
```

詳細モード

```
STW -u Admin -verbose -d MyDsn -s MyScript
```

以下は、詳細モードの出力例です。

```
STW.exe
Copyright (c) 1996-2011, Micro Focus IP Development Limited. All Rights Reserved

DSN                : STW-Scratch
Username            : admin
Password            :
Script(s)           : MyScript [Project3]

Append result      : False
Increment result   : False
Output file        : MyOutput.txt
Verbose Mode       : ON

Searching executable assets...

.NET Script ' MyScript' [Project3] found

Executing MyScript [Project3] ...

Script: MyScript [Project3], Status: Failed, Verifications: Total 5, Passed 0, Failed 5
```

出力ファイル

```
STW -outputfile C:¥Temp¥MyFile.txt
```

結果ディレクトリの指定

```
STW -s scriptname -resultdir C:¥Temp¥TestResults
```

または

```
STW -s scriptname -rd C:¥Temp¥TestResults
```

コマンドラインからの資産のインポートとエクスポート

Silk Test Workbench のインポートユーティリティ (STWImport.exe) およびエクスポートユーティリティ (STWExport.exe) をコマンドラインプロンプトから直接実行して、データベースの資産をインポートまたはエクスポートできます。

再利用可能な構成ファイルを作成して、繰り返し使用するパラメータ値を保存することもできます。




注: Silk Test Workbench 18.5 以降を使用している場合、18.5 より前のバージョンの Silk Test Workbench でエクスポートした資産をインポートできません。このような資産をインポートする場合は、**StwConvertExportedXML** プログラムを使用してください。詳細については、「[資産のインポート](#)」を参照してください。


資産のインポートのコマンド ライン パラメータ


STWImport.exe コマンド ライン ユーティリティを使用して、コマンド ラインから資産をインポートします。インポート パラメータは全部をコマンド ラインから指定することも、または特に記載のないかぎりは構成ファイルに指定することもできます。

ユーティリティは、¥Silk¥Silk Test¥ng¥gui¥ にあります。


コマンド ライン インポート パラメータ

構文	値	必須かどうか	説明
Username (-u)	ユーザー定義	はい	接続に使用されるユーザー名。コマンド ラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されません。Windows 認証を使用する場合は、ユーザー名は省略します。
Password (-p)	ユーザー定義	はい	接続に使用されるパスワード名。このパラメータはパスワードが Nothing 以外の場合に必要です。コマンド ラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されません。Windows 認証を使用する場合は、パスワードは省略します。
DSN (-d)	ユーザー定義	はい	データベースへの接続に使用する DSN の名前。このパラメータは、「SilkTest」以外の DSN 名の場合に必要です。コマンド ラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されません。
ImportFile (-if)	ユーザー定義		インポートする ZIP ファイルのファイル名または場所。
Execute (-ex)	Yes No (下記の注意を参照)	はい	インポート操作をコミットします。デフォルト値は、「No」です。
Filename (-f)	ユーザー定義	いいえ	コマンド ライン パラメータを格納する INI 構成ファイルです。コマンド ラインでのみ指定できます。  注: INI ファイル パラメータはコマンド ライン パラメータを上書きしますが、例外があります。INI ファイルで指定された Projects (-prj) および System (-s) パラメータの値は、コマンド ライン上に指定された同じパラメータを上書きしません。それぞれの値が両方とも使われます。
AllVersions (-av)	Yes No (下記の注意を参照)	いいえ	資産のすべてのバージョンをインポートします。デフォルト値は、「No」です。
ReportResult (-rr)	ユーザー定義	いいえ	作成される要約レポート ファイルの名前と場所が含まれています。デフォルトの場所は現在のディレクトリです。
SilentMode (-sm)	Yes No (下記の注意を参照)	いいえ	コンソール出力の表示を抑制します。
Projectmapping (-pm)	ユーザー定義	いいえ	インポート ファイルの元のプロジェクトの代わりに使用する対象のプロジェクトを指定します。縦線文字 () で区切って複数のマッピングを指定できます。

構文	値	必須かどうか	説明
Projects (-prj)	資産のノード	いいえ	<p>インポートする単一のプロジェクト資産またはプロジェクト資産のグループを指定します。このパラメータを指定しない場合、データベース内のすべての資産がインポートされます。ノードのグループは縦線文字 () で区切られます。形式は、<code>{ProjectName}.{AssetType}.{AssetName}.{Ver#}</code> です。例：</p> <pre style="background-color: #f0f0f0; padding: 5px;">-prj ProjectA.Script.SampleScript.1</pre> <p>縦線文字で区切る文字列は、二重引用符で囲む必要があります。例：</p> <pre style="background-color: #f0f0f0; padding: 5px;">-prj "Common ProjectB.Result"</pre> <p> 注: 資産の種類を指定する際は、単数形を使用してください。たとえば、「results」ではなく、「result」を使用します。資産名が 2 語で構成される場合 (visual test) は、スペースを削除してください (visualtest)。ただし、テストスクリプトだけは例外です。テストスクリプトを指定する際は、「script」という単語のみを使用してください。</p>
System (-s)	資産のノード	いいえ	<p>インポートする単一のシステム資産またはシステム資産のグループを指定します。このパラメータを指定しない場合、データベース内のすべてのシステム資産がインポートされます。ノードのグループは縦線文字 () で区切られます。形式は、<code>{SystemName}.{SystemType}</code>、<code>{SystemAssetName}</code> です。例：</p> <pre style="background-color: #f0f0f0; padding: 5px;">-s Options.Record.RecA</pre> <p>縦線文字で区切る文字列は、二重引用符で囲む必要があります。</p>
ProjectAssetResolve-Conflict (-parc)	s a p	いいえ	<p>データベースにすでに存在しているプロジェクト資産について、競合の解決方法を指定します。</p> <ul style="list-style-type: none"> • s を指定すると、インポートが中止されます。 • a を指定すると、インポート時に資産にバージョンが追加されます。 • p を指定した場合、競合する資産それぞれに対して、ユーザーに操作を選択するよう促すプロンプトを表示し、SilentMode パラメータを上書きします。デフォルトは p です。
SystemAssetResolve-Conflict (-sarc)	s p	いいえ	<p>データベースにすでに存在しているシステム資産について、競合の解決方法を指定します。</p> <ul style="list-style-type: none"> • s を指定すると、インポートが中止されます。 • p を指定した場合、競合する資産それぞれに対して、ユーザーに操作を選択するよう促すプロンプトを表示し、SilentMode パラメータを上書きします。デフォルトは「p」です。

 **注:** コマンドラインパラメータのデフォルト値を指定するには、コマンドでパラメータの指定を省略します。デフォルト以外の値を指定するには、値なしでパラメータだけを含めます。たとえば、資産のすべてのバージョンをインポートするには、インポートコマンドで AllVersions (-av) パラメータを値「Yes」なしで指定します。資産の最新バージョンだけをインポートするには、単にこのパラメータを省略するとデフォルト値の「No」が適用されます。

インポート構成ファイルでパラメータ値を指定するとき、パラメータまたはパラメータ値を省略すると、デフォルト値が使用されます。デフォルト以外の値の場合は、値を明示的に指定する必要があります。たとえば、資産のすべてのバージョンをインポートするには、AllVersions (-av) パラメータを値の「Yes」と共に指定します。

 **注:** Silk Test Workbench 18.5 以降を使用している場合、18.5 より前のバージョンの Silk Test Workbench でエクスポートした資産をインポートできません。このような資産をインポートする場合は、**StwConvertExportedXML** プログラムを使用してください。詳細については、「[資産のインポート](#)」を参照してください。

例

```
C:/>STWImport.exe -u Admin -p admin -d MyDBName -mt sf -mn "c:¥MySTAssetFiles" -prj ProjectA -pm "ProjectA=ProjectB" -rr "c:¥STImport.txt" -ex
```

上記の例では、MySTAssetFiles ディレクトリに含まれる ProjectA の資産ファイルの最新バージョンが、MYDBName データベースの ProjectB にインポートされ、要約レポートを含むファイルが作成されます。

構成ファイルの例

```
C:/>STWImport.exe -u Admin -p admin -d MyDBName -f "C:¥STImport.ini"
```

上記の例には、コマンドライン上で指定する必要のあるインポートパラメータだけが含まれています。その他のすべてのパラメータは構成ファイルに保存されます。

コマンドラインインポート構成ファイルの作成


設定を永続的に保存するには、コマンドラインインポート構成ファイルを作成します。インポートパラメータの大部分は、このファイルに保存できます。Username、Password、DSN、Filename などのいくつかのパラメータは、コマンドライン上で指定する必要があります。

1. オプションのインポートパラメータをすべて含めるための .INI ファイル (たとえば STImport.ini) を作成します。
2. ファイルを必要な場所に保存します。
3. 以下のインポートパラメータのリストをコピーして、構成ファイルに貼り付けます。

```
[SilkTestImport]
Execute=
ImportFile=
AllVersions=
ReportResult=
SilentMode=
Projects=
System=
ProjectAssetResolveConflict=
SystemAssetResolveConflict=
```

4. 各インポートパラメータの値を設定します。値は引用符で囲む必要はありません。値が設定されない場合、またはファイルでこのパラメータが省略された場合は、デフォルト値が使用されます。パラメータ値では大文字と小文字は区別されません。
5. ファイルを保存します。
6. コマンドラインからファイルにアクセスするには、Filename (-f) パラメータとファイルの場所を指定します。例：

```
C:¥>STWImport.exe -u Admin -p admin -d YourDBName -f "C:¥STImport.ini"
```


 **注:** 構成ファイルの見出し [SilkTestImport] は、ファイルで最初に指定されている必要がありますが、パラメータの順番は変更できます。パラメータをコメントアウトする必要がある場合、パラメータ名の前に円記号 (¥) を入力します。構成ファイルに指定されたパラメータは、コマンドライン上に指定されている同じパラメータを上書きしますが、例外が 1 つあります。構成ファイル内とコマンドライン上の両方に Projects または System パラメータが指定されている場合、それぞれの値が使用されます。


資産のエクスポートのコマンドラインパラメータ


STWExport.exe コマンドラインユーティリティを使用して、コマンドラインから資産をエクスポートします。エクスポートパラメータは全部をコマンドラインから指定することも、または特に記載のないかぎりは構成ファイルに指定することもできます。

ユーティリティは、¥Silk¥Silk Test¥ng¥gui¥ にあります。

コマンドラインエクスポートパラメータ

構文	値	必須かどうか	説明
Username (-u)	ユーザー定義	はい	接続に使用されるユーザー名。コマンドラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されます。Windows 認証を使用する場合は、ユーザー名は省略します。
Password (-p)	ユーザー定義	はい	接続に使用されるパスワード名。このパラメータはパスワードが Nothing 以外の場合に必要です。コマンドラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されます。Windows 認証を使用する場合は、パスワードは省略します。
DSN (-d)	ユーザー定義	はい	データベースへの接続に使用する DSN の名前。このパラメータは、「SilkTest」以外の DSN 名の場合に必要です。コマンドラインでのみ指定できます。このパラメータを構成ファイルで指定しても無視されます。
ExportFile (-ef)	ユーザー定義	はい	エクスポートする ZIP ファイルのファイル名または場所。
Execute (-ex)	Yes No (下記の注意を参照)	はい	エクスポート操作をコミットします。デフォルト値は、「No」です。
Filename (-f)	ユーザー定義	いいえ	コマンドラインパラメータを格納する INI 構成ファイルです。コマンドラインでのみ指定できます。  注: INI ファイルパラメータはコマンドラインパラメータを上書きしますが、例外があります。INI ファイルで指定された Projects (-prj) および System (-s) パラメータの値は、コマンドライン上に指定された同じパラメータを上書きしません。それぞれの値が両方とも使われます。
AllVersions (-av)	Yes No (下記の注意を参照)	いいえ	資産のすべてのバージョンをエクスポートします。デフォルト値は、「No」です。
ReportResult (-rr)	ユーザー定義	いいえ	作成される要約レポートファイルの名前と場所が含まれています。デフォルトの場所は現在のディレクトリです。

構文	値	必須かどうか	説明
SilentMode (-sm)	Yes No (下記の注意を参照)	いいえ	コンソール出力の表示を抑止します。
Projects (-prj)	資産のノード	いいえ	<p>エクスポートする単一のプロジェクト資産またはプロジェクト資産のグループを指定します。このパラメータを指定しない場合、データベース内のすべての資産がエクスポートされます。ノードのグループは縦線文字 () で区切られます。形式は、{ProjectName}、{AssetType}、{AssetName}、{Ver#} です。例：-prj ProjectA.Visualtest.SampleTest.1。縦線文字 () で区切る文字列は、二重引用符で囲む必要があります。例：-prj "Common ProjectB.Result"。</p> <p> 注: 資産の種類を指定する際は、単数形を使用してください。たとえば、「results」ではなく、「result」を使用します。資産名が 2 語で構成される場合 (visual test) は、スペースを削除してください (visualtest)。ただし、テストスクリプトだけは例外です。テストスクリプトを指定する際は、「script」という単語のみを使用してください。</p>
System (-s)	資産のノード	いいえ	<p>エクスポートする単一のプロジェクト資産またはプロジェクト資産のグループを指定します。このパラメータを指定しない場合、データベース内のすべてのシステム資産がエクスポートされます。ノードのグループは縦線文字 () で区切られます。形式は、{SystemName}、{SystemType}、{SystemAssetName} です。例：-s Options.Record.RecA。縦線文字 () で区切る文字列は、二重引用符で囲む必要があります。</p>
ExportAssociated (-ea)	Yes No (下記の注意を参照)	いいえ	データベースのすべてのプロジェクトから、参照されるすべての資産がエクスポートされるようにします。デフォルト値は、「No」です。
OverwriteFile (-of)	Yes No (下記の注意を参照)	いいえ	エクスポート時にファイル名が競合した場合、既存のファイル (読み取り専用ファイルを含む) を上書きします。デフォルト値は、「Yes」です。

 **注:** コマンドラインパラメータのデフォルト値を指定するには、エクスポート コマンドでパラメータの指定を省略します。デフォルト以外の値を指定するには、値なしでパラメータだけを含めます。たとえば、資産のすべてのバージョンをエクスポートするには、エクスポート コマンドで AllVersions (-av) パラメータを値「Yes」なしで指定します。資産の最新バージョンだけをエクスポートするには、単にこのパラメータを省略するとデフォルト値の「No」が適用されます。

エクスポート構成ファイルでパラメータ値を指定するとき、パラメータまたはパラメータ値を省略すると、デフォルト値が使用されます。デフォルト以外の値の場合は、値を明示的に指定する必要があります。たとえば、資産のすべてのバージョンをエクスポートするには、AllVersions (-av) パラメータを値「Yes」付きで指定します。

例

```
C:/>STWExport.exe -u Admin -p admin -d MyDBName -mn "c:¥MySTAssetFiles" -mt ds -rr "c:¥STExport.txt" -prj ProjectA -ex
```

上記の例では、ProjectA の資産ファイルの最新バージョンが C:¥MySTAssetFiles にエクスポートされ、結果の要約レポートを含むファイルが作成されます。

構成ファイルの例

```
C:¥>STWExport.exe -u Admin -p admin -d YourDBName -f "C:¥STExport.ini"
```

コマンドラインエクスポート構成ファイルの作成


エクスポート設定を永続的に保存するには、コマンドラインエクスポート構成ファイルを作成します。エクスポートパラメータの大部分は、このファイルに保存できます。Username、Password、DSN、Filenameなどのいくつかのパラメータは、コマンドライン上で指定する必要があります。

1. オプションのエクスポートパラメータをすべて含めるための .INI ファイル (たとえば STExport.ini) を作成します。
2. ファイルを必要な場所に保存します。
3. 以下のエクスポートパラメータのリストをコピーして、構成ファイルに貼り付けます。

```
[SilkTestExport]
Execute=
ExportFile=
AllVersions=
ReportResult=
SilentMode=
Projects=
System=
ExportAssociated=
OverwriteFile=
```

4. 各エクスポートパラメータの値を設定します。値は引用符で囲む必要はありません。値が設定されていない場合は、デフォルトが使用されます。パラメータ値では大文字と小文字は区別されません。
5. ファイルを保存します。
6. コマンドラインからファイルにアクセスするには、Filename (-f) パラメータとファイルの場所を指定します。例：

```
C:¥>STWExport.exe -u Admin -p admin -d YourDBName -f "C:¥STExport.ini"
```

 **注:** 構成ファイルの見出し [SilkTestExport] は、ファイルで最初に指定されている必要がありますが、パラメータの順番は変更できます。パラメータをコメントアウトする必要がある場合、パラメータ名の前に円記号 (¥) を入力します。構成ファイルに指定されたパラメータは、コマンドライン上に指定されている同じパラメータを上書きしますが、例外が 1 つあります。構成ファイル内とコマンドライン上の両方に Projects または System パラメータが指定されている場合、それぞれの値が使用されます。

ログ

コマンドラインからの実行中に、Silk Test Workbench は、%LOCALAPPDATA%¥silk¥silktest¥logs ディレクトリにある .log ファイルにイベントを記録します。デフォルトでは、ログレベルは情報モードに設定されており、コマンドラインからの実行ごとに、PID と日時を使った名前の新しいログファイルが作成されます。同じプログラムを複数回実行すると、古いファイルは削除され、新しいファイルが作成されます。

次の表には、コマンドライン実行可能ファイルに対応するファイル名のサンプルを示します。

プログラム	サンプルファイル名
SilkTest.exe	SilkTest(Pid=25184)_20171030_1635.log
STW.exe	STW(Pid=13204)_20171031_0839.log

プログラム	サンプル ファイル名
STWDBConfig.exe	STWDBConfig(Pid=32924)_20171030_1635.log
STWExport.exe	STWExport(Pid=32560)_20171030_1635.log
STWImport.exe	STWImport(Pid=26640)_20171030_1635.log
STWMaint.exe	STWMaint(Pid=13488)_20171030_1618.log
STWPurge.exe	STWPurge(Pid=30096)_20171030_1635.log

ログの設定は、Silk Test Workbench インストールの gui フォルダにある workbench.properties ファイルを変更して行います。デフォルトでは、このフォルダは C:\Program Files (x86)\Silk\SilkTest\ng\gui になります。このファイルでは、ログ レベル、ログ形式、ログ ディレクトリなどを指定できます。

たとえば、1 つのプログラムに対して複数のログ ファイルを作成する必要がある場合は、nativeLog.clearPerProcessLogs を 0 に設定すると、ファイルが削除されなくなります。

また、nativeLog.LogPerProcess を 0 に設定すると、ファイルの作成方法を変更できます。この場合、すべての実行可能ファイルに対して、silktest.log という名前のファイルにログが記録されるようになります。



注: nativeLog.OverWriteExisting を 0 に設定しない限り、実行するたびにファイルが作成しなおされます。

Silk Test Workbench のオプション

テスト環境とテスト方法を反映するには、デフォルト オプションを変更します。オプションを変更するには、**ツール > オプション** をクリックします。

グローバル オプション


すべてのプロジェクトのプロパティを設定するには、**グローバル** オプションを使用します。

グローバル オプションは、以下のとおりです。


資産管理	デフォルトの保存動作	ツールバーのボタン、資産のダイアログ ボックス、またはショートカット キーの組み合わせ (Ctrl+S) で資産を保存するときの、デフォルトの保存動作を設定します。動作は、資産の新しいバージョンを作成する、または資産の現在のバージョンを置き換える、のいずれかから選択します。
	資産のバージョンの最大数	保持する資産のバージョンの最大数。0 を指定すると、すべてのバージョンが保持されます。バージョンの最大数に到達すると、資産の新しいバージョンが保存され、最も古いバージョンが削除されます。資産をインポートする場合、バージョンの数が最大数を超えていたとしても、すべてのバージョンがインポートされます。このようにしてインポートされた資産に対して新しいバージョンが保存されるとき、指定した資産の最大数と一致するように、最も古いバージョンの資産から削除されます。
	ページの実行に必要なアクセス権	プロジェクトの資産をパーズするために最低限必要なユーザー アクセス権を設定します。デフォルト値は、 スクリプト作成者 です。 フル アクセス も選択できます。このオプションは、管理者権を持つ Silk Test Workbench ユーザーだけが変更できます。現在のユーザーがパーズを実行できるアクセス権を持つプロジェクトがない場合、Silk Test Workbench メニューの ファイル > 資産のバージョンをパーズ... 項目は無効になります。それ以外の場合、 資産のバージョンをパーズ ダイアログの プロジェクト ツリーに、ユーザーが資産をパーズできるプロジェクトが表示されます。
	[資産のバージョンをパーズ] のプロジェクトの選択状態を保持	はい に設定すると、 資産のバージョンをパーズ ダイアログ ボックスで、前回実行時のプロジェクトの選択状態が保持されます。 いいえ に設定すると、ダイアログ ボックスを開いたときに、プロジェクトは選択されません。

eCATT

ツール > オプション > グローバル > 統合

 **注:** これらのオプションは、Silk Test Workbench が eCATT と統合されている場合にのみ利用できません。

RFC のトレース Silk Test Workbench の RFC 呼び出しの診断トレースを有効にするには **オン** に設定します。診断トレースは、**トレース ファイル** に保存されます。

 **注:** RFC は、librfc32.dll という SAP .dll に実装されている SAP API です。トレース ファイルはこの .dll 内のコードによって作成され、トレース ファイルの保存場所もこのコードによって設定されます。デフォルトの場所はシステム フォルダです。

RFC トレース ファイルをダンプするパスの Windows 環境変数を設定します。変数は `RFC_TRACE_DIR` で、その値はパスにする必要があります。これについては、SAP が提供している RFCSDK のヘルプで説明されています。

全般オプション

ツール > オプション > 全般

全般オプションは、以下のとおりです。

デフォルトの結果表示	Silk Test Workbench によって 結果 ウィンドウに表示されるタブを選択します。
デフォルトの待機タイムアウト (ミリ秒)	Silk Test Workbench がオブジェクトを待機する期間を指定します。デフォルトは 5000 です。
記録時にフラッシュする	テストの記録中に、Silk Test Workbench がタスクバーの Silk Test Workbench ウィンドウをフラッシュするかどうかを指定します。
検索時に最小化	検索 ボタンをクリックしたときに、Silk Test Workbench が Silk Test Workbench ウィンドウを最小化するかどうかを指定します。
再生時に最小化	テストの再生中に、Silk Test Workbench が Silk Test Workbench ウィンドウを最小化するかどうかを指定します。
記録時に最小化	テストの記録中に、Silk Test Workbench が Silk Test Workbench ウィンドウを最小化するかどうかを指定します。
結果を自動的に開く	テストの再生後に Silk Test Workbench が結果を自動的に表示するかどうかを指定します。
再生時にすべて保存	テストの再生中に、Silk Test Workbench がすべてのファイルを保存するかどうかを指定します。
ブラウザ構成ダイアログの表示	資産にブラウザ構成がある場合にブラウザを選択できるように、Silk Test Workbench が ブラウザ構成 ダイアログを表示するかどうかを指定します。この値を いいえ に設定すると、Silk Test Workbench はアプリケーション構成で指定したブラウザの種類を使用します。
再生中にローカル変数ウィンドウを表示する	再生中に Silk Test Workbench が ローカル変数 ウィンドウを表示し、再生終了時に非表示にするかどうかを指定します。
[テスト ロジック デザイナ] の [ロジック ステップ] ページを表示する	Silk Test Workbench が テスト ロジック デザイナ ウィザードに ロジック ステップ ペインを表示するかどうかを指定します。ウィザードの ロジックのステップ ペインのタイトルの横にある矢印ボタンをクリックして、このペインの表示/非表示を切り替えることもできます。
モバイル構成ダイアログの表示	モバイル デバイス アプリケーション構成を使用した記録、再生、または、ビジュアル テストまたは .NET スクリプトの識別中に モバイル デバイス構成 ダイアログを表示するかどうかを指定します。
[再生完了] ダイアログの表示	テスト再生が終了したときに、Silk Test Workbench が 再生完了 ダイアログ ボックスを表示するかどうかを指定します。
[再生] ダイアログの表示	テストを開始する前に、Silk Test Workbench が 再生 ダイアログ ボックスを表示するかどうかを指定します。

[記録完了] ダイアログの表示	Silk Test Workbench が 記録完了 ダイアログ ボックスを表示するかどうかを指定します。
[状態の記録] ダイアログの表示	記録を開始するときに、Silk Test Workbench が 状態の記録 ダイアログ ボックスを表示するかどうかを指定します。
[テスト ロジック デザイナ] の [要約] ページを表示する	Silk Test Workbench が テスト ロジック デザイナ ウィザードに 要約 ペインを表示するかどうかを指定します。この設定は、 テスト ロジック デザイナ ウィザードの 要約 ペインで変更することもできます。
ウィザードの [ようこそ] 画面の表示	Silk Test Workbench のウィザードごとに ようこそ ページを表示するかどうかを指定します。この設定は、各ウィザードの ようこそ ページで変更することもできます。

全般オプションの変更

全般 オプションは、起動やビジュアル テストまたはスクリプトの記録および再生を行う際の全般的な動作を制御します。


1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーで **全般** をクリックします。全般オプションが右側のパネルに表示されます。
3. 最初の列でオプションの名前を選択し、2 番目の列で新しい値を選択または入力して、オプションを変更します。
オプションを選択すると、簡単な説明が下のパネルに表示されます。
4. Common プロジェクトに追加するすべての資産に接頭辞 *Common* を追加するには、**Common プロジェクト資産に接頭辞を追加する** リスト ボックスから **資産名と 'Common' を使用する** を選択します。
5. **デフォルトの結果表示** リスト ボックスから、**結果** ウィンドウに表示するタブを選択します。
6. Silk Test Workbench がオブジェクトを待機する時間を指定するには、使用する時間を **デフォルトの待機タイムアウト (ミリ秒)** テキスト ボックスに入力します。
7. テストの記録中にタスクバーの Silk Test Workbench ウィンドウをフラッシュするには、**記録時にフラッシュする** リスト ボックスから **はい** を選択します。
8. **検索** ボタンをクリックしたときに Silk Test Workbench ウィンドウを最小化するには、**検索時に最小化** リスト ボックスから **はい** を選択します。
9. テストの再生時に Silk Test Workbench ウィンドウを最小化するには、**再生時に最小化** リスト ボックスから **はい** を選択します。
10. テストの記録時に Silk Test Workbench ウィンドウを最小化するには、**記録時に最小化** リスト ボックスから **はい** を選択します。
11. テストの再生後に結果を自動的に表示するには、**結果を自動的に開く** リスト ボックスから **はい** を選択します。
12. テストの再生時にすべてのファイルを保存するには、**再生時にすべて保存** リスト ボックスから **はい** を選択します。
13. 資産にブラウザ構成があるときに、ブラウザを選択するために、**ブラウザ構成** ダイアログを開くには、**ブラウザ構成ダイアログの表示** リスト ボックスから **はい** を選択します。
14. 再生中に **ローカル変数** ウィンドウを表示して、再生が終わったら非表示にするには、**ローカル変数ウィンドウの表示** リスト ボックスから **はい** を選択します。
15. **テスト ロジック デザイナ** ウィザードで **ロジック ステップ** ペインを表示するには、**[テスト ロジック デザイナ] の [ロジック ステップ] ページを表示する** リスト ボックスから **はい** を選択します。
ウィザードの **ロジックのステップ** ペインのタイトルの横にある矢印ボタンをクリックして、このペインの表示/非表示を切り替えることもできます。
16. テスト再生の終了時に **再生完了** ダイアログを表示するには、**[再生完了] ダイアログの表示** リスト ボックスから **はい** を選択します。

- 17テストの開始前に **再生** ダイアログを表示するには、**[再生] ダイアログの表示** リスト ボックスから **はい** を選択します。
再生 ダイアログ ボックスでは、結果名、説明、およびオプション ファイルの詳細を割り当てることができます。
- 18テストの開始前に **記録完了** ダイアログを表示するには、**[記録完了] ダイアログの表示** リスト ボックスから **はい** を選択します。
記録完了 ダイアログ ボックスでは、テストの再生、ビジュアル テストへの移動、またはテストの保存を行うことができます。
- 19記録の開始時に **状態の記録** ダイアログ ボックスを表示するには、**[状態の記録] ダイアログの表示** リスト ボックスから **はい** を選択します。
- 20テスト **ロジック デザイナ** ウィザードで **要約** ペインを表示するには、**[テスト ロジック デザイナ] の [要約] ページを表示する** リスト ボックスから **はい** を選択します。
この設定は、**テスト ロジック デザイナ** ウィザードの **要約** ページで変更することもできます。
- 21各ウィザードで [ようこそ] ページを表示するには、**ウィザードの [ようこそ] 画面の表示** リスト ボックスから **はい** を選択します。
この設定は、各ウィザードの **ようこそ** 画面で変更することもできます。
- 22OK をクリックします。

開始画面オプション


ツール > オプション をクリックしてから **開始画面** をクリックします。

このオプションを使用し、**開始画面** の設定を制御します。

 **注:** いつでも Ctrl+Alt+S を押すか、または **表示 > 開始画面** をクリックして、開始画面を表示できます。

Silk Test Workbench の開始時に開始画面を表示する	Silk Test Workbench の開始時に 開始画面 を表示するには、 はい に設定します。
[最近使った資産] に表示する数	最近使った資産 リストに表示する項目の数を選択します。

フラグ 割り当て先	選択されたユーザーに割り当てられているフラグを表示するには、このオプションを使用します。現在のユーザーに割り当てられているフラグを表示するには、 現在のユーザー に設定します。プロジェクト管理者に割り当てられているフラグを表示するには、 Admin に設定します。このオプションを Admin に設定するには、管理者権限が必要です。
------------------	---

 **注:** ユーザーがシステムから削除されると、そのユーザーの名前は **割り当て先** リストから削除され、そのユーザーに割り当てられていたフラグはプロジェクト管理者に再割り当てされます。

最終更新者	すべてのユーザーによって変更されたフラグを表示するか、または特定のユーザーによって変更されたフラグのみを表示するには、このオプションを設定します。現在のデータベースでビジュアル テストに対してフラグを最後に入力または変更したユーザーがリストに表示されます。
--------------	--

日付順	更新日時が最も古いフラグをリストの最上部に表示するには、 昇順 に設定します。更新日時が最も新しいフラグをリストの最上部に表示するには、 降順 に設定します。
------------	---

プロジェクト	選択されたプロジェクトで作成されたフラグのみを表示するには、このオプションを使用します。デフォルト設定では、すべてのプロジェクトのフラグが表示されるようになっています。Common プロジェクトまたは別のプロジェクトをリストから選択します。
---------------	--

作成したフラグの表示 ビジュアルテストで作成されたフラグ、結果で作成されたフラグ、またはビジュアルテストと結果の両方で作成されたフラグを表示するよう、リストからオプションを選択します。

開始画面オプションの変更

開始画面 オプションを使用して、Silk Test Workbench が起動したときに **開始画面** を非表示にします。また、**開始画面** の **フラグ** ペインで、フラグの表示方法を指定することもできます。また、いつでも Ctrl+Alt+S を押すか、または **表示** > **開始画面** を選択して、開始画面を表示できます。

1. ツール > **オプション** をクリックします。
2. **オプション** メニュー ツリーの **開始画面** の隣にあるプラス記号 (+) をクリックします。**開始画面** オプションが右側のパネルに表示され、**フラグ** サブカテゴリが **開始画面** の下のアウトライン ビューに表示されます。**Silk Test Workbench** の開始時に**開始画面を表示する** を **はい** に設定して、Silk Test Workbench の開始時に **開始画面** を表示できます。**開始画面** では、ユーザーは、ビジュアルテストを記録し、開いてクイックテストを実行する、管理タスク、テストの記録タスク、編集タスクを実行する、フラグを使用して、プロジェクト協業を実施する、アプリケーション テストを迅速に作成するためのアシスタンスを利用するといった作業が可能です。
3. アウトライン ビューで、**フラグ** をクリックしてオプションを設定します。
4. 環境に適したオプションを設定します。

割り当て先 選択されたユーザーに割り当てられているフラグを表示するには、このオプションを使用します。現在のユーザーに割り当てられているフラグを表示するには、**現在のユーザー** に設定します。プロジェクト管理者に割り当てられているフラグを表示するには、**Admin** に設定します。このオプションを **Admin** に設定するには、管理者権限が必要です。



注: ユーザーがシステムから削除されると、そのユーザーの名前は **割り当て先** リストから削除され、そのユーザーに割り当てられていたフラグはプロジェクト管理者に再割り当てされます。

最終更新者 すべてのユーザーによって変更されたフラグを表示するか、または特定のユーザーによって変更されたフラグのみを表示するには、このオプションを設定します。現在のデータベースでビジュアルテストに対してフラグを最後に入力または変更したユーザーがリストに表示されます。

日付順 更新日時が最も古いフラグをリストの最上部に表示するには、**昇順** に設定します。更新日時が最も新しいフラグをリストの最上部に表示するには、**降順** に設定します。

プロジェクト 選択されたプロジェクトで作成されたフラグのみを表示するには、このオプションを使用します。デフォルト設定では、すべてのプロジェクトのフラグが表示されるようになっています。Common プロジェクトまたは別のプロジェクトをリストから選択します。

作成したフラグの表示 ビジュアルテストで作成されたフラグ、結果で作成されたフラグ、またはビジュアルテストと結果の両方で作成されたフラグを表示するよう、リストからオプションを選択します。

5. **OK** をクリックします。

記録オプション

記録オプションは、以下のカテゴリに分けられます。

全般	資産とマウスの移動の設定、オブジェクト マップの場所、無視するクラス、および、記録時にテスト対象アプリケーション (AUT) のサイズを変更して、AUT の隣に Silk Recorder を表示するかを設定します。
ロケーター	記録を一時停止するためのホット キーの組み合わせ、およびロケーターの記録に使用するモードを指定します。
出力	記録中にコントロールをキャプチャするかどうか、プロパティを記録するかどうか、画面のキャプチャ方法、個別の画面をキャプチャするかどうかを指定します。
ホット キー	どのキーの組み合わせが記録中に検証を挿入するか、または記録を開始および停止するかを指定します。
xBrowser	カスタム属性、記録中に無視するブラウザ属性、DOM 関数の代わりに、ユーザーの入力そのものを記録するかどうかを指定します。
WPF	カスタム属性、および記録や再生の対象にする WPF クラスの名前を指定します。
Rumba	画面検証を作成するときに無視する Rumba 画面上のオブジェクトを指定します。... をクリックして、 除外オブジェクトの編集 ダイアログ ボックスを開きます。最初の行にある + を押すと、新しい除外オブジェクトが追加されます。各オブジェクトの行 ("オブジェクト 1"、"オブジェクト 2" など) にある x ボタンを使用すると、個々の除外オブジェクトを削除できます。この設定では、画面検証が記録されるたびに、設定内の除外オブジェクトがその画面検証に設定されます。個々の画面検証内の除外オブジェクトは手動で編集できますが、これによって、(必要に応じて、個々の検証に合わせて除外オブジェクトが調整されるようにして) 設定が更新されることはありません。
JavaSWT	ロケーター生成中に使用するカスタム属性を指定します。
SAP	ロケーター生成中に使用するカスタム属性を指定します。
Swing	ロケーター生成中に使用するカスタム属性を指定します。
Win32	ロケーター生成中に使用するカスタム属性を指定します。
WinForms	ロケーター生成中に使用するカスタム属性を指定します。
Open Agent の設定	エージェントに直接送信されるカスタム設定を行います。サポートに勧められた場合など、非常に特殊なテスト シナリオでのみ設定されるべきです。

イメージ資産記録設定の設定

イメージのクリックまたは画面検証を記録するかどうか、およびイメージ資産の精度レベルにデフォルト値を設定するかどうかを指定します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **全般** をクリックします。
4. **イメージ資産のデフォルト精度レベル** リスト ボックスから、1 (低精度) から 10 (高精度) までの値を選択し、新しいイメージ資産のデフォルト精度レベルを設定します。
5. オブジェクト クリックを記録するには、**'ImageClick' を記録する** リスト ボックスから **はい** を選択します。
6. 画面検証を記録するには、**画面検証を記録する** リスト ボックスから **はい** を選択します。
7. **OK** をクリックします。

オブジェクト マップの詳細設定の設定

オブジェクト マップを記録するかどうか、オブジェクト マップを記録するプロジェクト、ロックされたオブジェクト マップの処理方法を指定します。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの 記録 の隣にあるプラス記号 (+) をクリックします。 記録 オプション が右側のパネルに表示されます。
3. ロケーター をクリックします。
4. オブジェクト マップ エントリを記録するか、XPath ロケーターを記録するかを定義するには、 **オブジェクト マップ エントリを記録する** リストから適切な記録モードを選択します。
 - **オブジェクト マップ エントリ (新しいオブジェクトと既存のオブジェクト)**。これはデフォルトのモードです。
 - **XPath ロケーター (新しいオブジェクトと既存のオブジェクト)**。
 - **XPath ロケーター (新しいオブジェクトのみ)**。オブジェクト マップに既に存在するオブジェクトに対しては、オブジェクト マップ エントリが再利用されます。この設定を選択すると、AUT のメインコントロールに対するオブジェクト マップを作成し、AUT に対して追加のテストを作成する間、これらのオブジェクト マップを保持することができます。
5. オプション メニュー ツリーで **全般** をクリックします。
6. ロックされたオブジェクト マップのロケーターを記録するには、 **ロック済みのオブジェクト マップに対してロケーターを記録する** リスト ボックスから **はい** を選択します。

はい を選択した場合、オブジェクト マップがロックされていると、以前に識別されたオブジェクト マップ項目が適用可能な場合に使用され、新しいオブジェクトを記録する場合にはロケーターが使用されます。

いいえ を選択した場合、記録する前に、オブジェクト マップがロック解除されるのを待機する必要があります。
7. Silk Test Workbench がオブジェクト マップ項目を検索し、一致した項目が見つからない場合に項目を作成するプロジェクトを指定するには、 **オブジェクト マップの場所** リスト ボックスからオプションを選択します。

デフォルトでは、Silk Test Workbench は現在のプロジェクトを検索して、記録時に使用する一致したオブジェクト マップ項目が決まります。一致する項目が見つからない場合、Silk Test Workbench はすべての参照プロジェクトを検索します。一致する項目が参照プロジェクトで見つかり、その項目が使用されます。一致する項目が参照プロジェクトで見つからない場合、Silk Test Workbench によって、現在のプロジェクトに新しい項目が作成されます。

現在のプロジェクト を選択した場合、オブジェクト マップ項目は、ビジュアル テストまたはスクリプトが格納されているプロジェクトを使用します。
8. **OK** をクリックします。

マウス移動の詳細設定

マウス移動イベントを使用する Web アプリケーション、Win32 アプリケーション、および Windows Forms アプリケーションでマウス移動操作を記録するかどうかを指定します。たとえば、Apache Flex や Swing など、xBrowser テクノロジ ドメインの子ドメインのマウス移動イベントを記録することはできません。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの 記録 の隣にあるプラス記号 (+) をクリックします。 記録 オプション が右側のパネルに表示されます。
3. **全般** をクリックします。
4. マウス移動操作を記録するには、 **マウスの移動操作を記録する** リスト ボックスから **はい** を選択します。

マウスの移動イベントを使用する Web ページをテストする場合は、 **はい** を選択します。

Silk Test Workbench では、スクリプトを短くするために、マウスが置かれた要素またはその親が変化するマウスの移動イベントのみが記録されます。
5. マウスの移動操作を記録する場合、MoveMouse 操作が記録される前に、どのくらいの間マウスが不動状態になければならないかを、 **マウスの移動記録遅延** テキスト ボックスにミリ秒単位で指定します。

デフォルト値は、200 に設定されています。

マウスの移動操作は、この時間、マウスが静止している場合にのみ記録されます。遅延を短くすると、予期しないマウスの移動操作が増加します。遅延を長くすると、操作を記録するためにマウスを静止しておく必要があります。

6. **OK** をクリックします。

無視するクラスの設定

オブジェクト階層を単純化し、テストスクリプトや関数のコードの行の長さを短くするために、次のテクノロジーの確実に不要なクラスに対するコントロールを抑制できます。

- Win32
- Java AWT/Swing
- Java SWT/Eclipse
- Windows Presentation Foundation (WPF)

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **全般** をクリックします。
4. **無視するクラス** グリッドで、記録や再生中に無視するクラスの名前を入力します。
複数のクラス名を指定する場合にはカンマで区切ります。
5. **OK** をクリックします。

ロケータの詳細設定の設定

記録を一時停止するためのホット キーの組み合わせ、およびロケータの記録に使用するモードを指定します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **ロケータ** をクリックします。
4. 記録を一時停止するために使用するホット キーの組み合わせを設定するには、**ロケータ識別ホットキー** リスト ボックスからキーの組み合わせを選択します。
デフォルトのホット キーの組み合わせは、Ctrl+Shift です。ただし、Silk Test Classic のユーザーは Ctrl+Alt を使用することを好む場合があります。
5. **ロケータ識別モード** リスト ボックスから、ロケータの記録に使用するモードを選択します。
 - **クリック** : 記録するオブジェクトをクリックで識別するには、このモードを使用します。デフォルトでは、最適なパフォーマンスを提供するこのモードが、Silk Test Workbench がロケータをキャプチャする際に使用されます。ただし、ロケータをキャプチャするときにアプリケーションは停止しません。この結果、このモードはテスト対象アプリケーションの画面の変化が急激でない場合に最適に機能します。たとえば、複数のオブジェクトが同時に表示される Web サイトをテストしている場合には、ホットキー モードを使用してテストを記録します。
 - **ホット キー** : 記録するオブジェクトをホット キーの組み合わせを押して識別するには、このモードを使用します。このモードでは、ロケータをキャプチャするときにアプリケーションは一時停止します。たとえば、複数のウィンドウを開いたり閉じたりする Web サイトをテストしている場合には、正しいウィンドウをキャプチャすることを確実にするためにアプリケーションを一時停止するこのモードを使用すると良いでしょう。デフォルトのホット キーの組み合わせは、Ctrl+Shift です。
6. オブジェクト マップ エントリを記録するか、XPath ロケータを記録するかを定義するには、**オブジェクト マップ エントリを記録する** リストから適切な記録モードを選択します。
 - **オブジェクト マップ エントリ (新しいオブジェクトと既存のオブジェクト)**。これはデフォルトのモードです。

- **XPath ロケーター (新しいオブジェクトと既存のオブジェクト)**。オブジェクト マップ エントリは記録されません。
- **XPath ロケーター (新しいオブジェクトのみ)**。オブジェクト マップに既に存在するオブジェクトに対しては、オブジェクト マップ エントリが再利用されます。この設定を選択すると、AUT のメインコントロールに対するオブジェクト マップを作成し、AUT に対して追加のテストを作成する間、これらのオブジェクト マップを保持することができます。

7. **OK** をクリックします。

UI オートメーション オプションの設定

Microsoft UI オートメーション (MSUIA) サポートの有効/無効の切り替え、ロケーター生成用のカスタム属性の指定、ロケーターから除外する属性の指定を行います。

UI オートメーション サポートを有効にして、ロケーターから除外する属性と値を指定することで、UI オートメーション プロバイダー インターフェイスを実装するアプリケーション (JavaFX や QT ベースのアプリケーションなど) のオブジェクトを、より正確に Silk Test Workbench が識別できるようにすることができます。

Silk Test Workbench には、ロケーターが記録時に一意となり、メンテナンスが容易になるようにする、高度なロケーター生成メカニズムが備えられています。使用するアプリケーションやフレームワークに応じて、最適な結果を得るためにデフォルト設定を変更できます。それぞれのテクノロジーで使用できる任意のプロパティ (整数や倍精度の数値、文字列、項目識別子、列挙値) を、カスタム属性として使用できます。

また、UI オートメーション プロバイダー インターフェイスを実装するアプリケーションの任意のプロパティをカスタム属性として設定して使用することもできます。最適な結果を得るために、テストで利用する要素にカスタム オートメーション ID を追加します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。 **再生** のサブカテゴリが表示されます。
3. **UI オートメーション** をクリックします。
4. 通常の Win32 コントロールの認識の代わりに Microsoft UI オートメーション サポートも有効化する場合は、**Microsoft UI オートメーション サポートの有効化** を **はい** に設定します。



注: Java FX アプリケーションをテストする場合は、UI オートメーション サポートをオンにする必要はありません。Silk Test Workbench では、Java FX アプリケーションに対して、デフォルトでオンになります。



注: UI オートメーション サポートは、標準のテクノロジー ドメイン固有のサポートをオーバーライドします。UI オートメーション サポートが必要なコントロールに対する操作を完了したら、UI オートメーション サポートをオフにして、標準コントロールに対する操作に戻すこともできます。

5. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
6. **UI オートメーション** をクリックします。
7. UI オートメーション プロバイダー インターフェイスを実装するアプリケーションのカスタム属性を追加する場合は、**カスタム属性** テキスト ボックスに使用する属性を入力します。
カスタム属性を使用すると、caption や index のような他の属性よりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は定義済みのウィジェットより前に他のオブジェクトが追加されると変更される可能性があるためです。
8. **ロケーター属性名除外リスト** グリッドで、記録中に無視する属性名を入力します。
たとえば、automationid という名前の属性を持つコントロールが AUT に複数存在するため無視したい場合は、automationid 属性名をリストに追加します。
複数の属性名を指定する場合にはカンマで区切ります。
9. **ロケーター属性値除外リスト** グリッドで、記録中に無視する属性値を入力します。
たとえば、すべての JavaFX コントロールに JavaFX<番号> という属性値があるため、デフォルトで値 JavaFX* がリストに追加されています。

複数の属性名を指定する場合にはカンマで区切ります。

100K をクリックします。

ブラウザの記録オプションの設定

カスタム属性、記録中に無視するブラウザ属性、DOM 関数の代わりに、ユーザーの入力そのものを記録するかどうかを指定します。

Silk Test Workbench には、ロケーターが記録時に一意となり、メンテナンスが容易になるようにする、高度なロケーター生成メカニズムが備えられています。使用するアプリケーションやフレームワークに応じて、最適な結果を得るためにデフォルト設定を変更できます。それぞれのテクノロジーで使用できる任意のプロパティ (整数や倍精度の数値、文字列、項目識別子、列挙値) を、カスタム属性として使用できます。

xBrowser アプリケーションでは、任意のプロパティを取得し、カスタム属性として使用することもできます。最適な結果を得るために、テストで利用する要素にカスタム オートメーション ID を追加します。


1. ツール > オプション をクリックします。

2. オプション メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。記録 オプション が右側のパネルに表示されます。


3. xBrowser をクリックします。

4. Web アプリケーションのカスタム属性を追加するには、**カスタム属性** テキスト ボックスに、使用する属性を入力します。

カスタム属性を使用すると、caption や index のような他の属性よりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は定義済みのウィジェットより前に他のオブジェクトが追加されると変更される可能性があるためです。

 **注:** Web アプリケーションにカスタム属性を含めるためには、HTML タグとして追加します。たとえば、MyAutomationID という属性を追加するには、`<input type='button' MyAutomationID='abc' value='click me' />` と入力します。

複数のオブジェクトに同じカスタム属性の値が割り当てられた場合は、そのカスタム属性を呼び出したときにその値を持つすべてのオブジェクトが返されます。たとえば、一意の ID として loginName を 2 つの異なるテキスト フィールドに割り当てた場合は、loginName 属性を呼び出したときに、両方のフィールドが返されます。

 **注:** 属性名の長さは、62 文字までという制限があります。

5. **ロケーター属性名除外リスト** テキスト ボックスで、記録中に無視する属性名を入力します。

このリストを使用して、サイズ、幅、高さ、スタイルなどの頻繁に変更される属性を指定します。ワイルドカード '*' および '?' を **ロケーター属性名除外リスト** で使用できます。

たとえば、height という名前の属性を記録しない場合には、height 属性名をリストに追加します。

複数の属性名を指定する場合にはカンマで区切ります。

6. **ロケーター属性値除外リスト** テキスト ボックスで、記録中に無視する属性値を入力します。

たとえば、x-auto という値を持つ属性を記録しない場合には、x-auto 属性値をリストに追加します。

一部の AJAX フレームワークでは、ページが再読み込みされるたびに変わる属性値が生成されます。このリストを使用して、そのような値を無視します。このリストでワイルドカードを使用することもできます。

複数の属性名を指定する場合にはカンマで区切ります。

7. DOM 関数の代わりにユーザーの入力そのものを記録するには、**ネイティブなユーザー入力を記録する** リスト ボックスから、**はい** を選択します。

たとえば、DomClick の代わりに Click を記録し、SetText の代わりに TypeKeys を記録するには、**はい** を選択します。

アプリケーションでプラグインまたは AJAX を使用している場合は、**はい** を指定して、ユーザーの入力そのものを使用します。アプリケーションでプラグインまたは AJAX を使用していない場合は、再生中にブラウザにフォーカスを設定したりブラウザをアクティブにしたりする必要がない高レベル DOM 関

数を使用することをお勧めします。テストで DOM 関数を使用すると、より高速になり、信頼性も高まります。

8. ブラウザの iframe および frame サポートを有効化するには、**iframe サポートの有効化** リストで **はい** を選択します。

Web アプリケーションの iframe の内容を無視して良い場合、iframe のサポートを無効にすると再生速度が速くなる可能性があります。たとえば、多くの広告が表示される Web ページやモバイル ブラウザでテストする際に iframe サポートを無効にすると、再生パフォーマンスが大幅に向上する可能性があります。このオプションは、Internet Explorer では無視されます。このオプションは、デフォルトで有効になっています。

9. **OK** をクリックします。

カスタム属性の設定

Silk Test Workbench には、ロケーターが記録時に一意となり、メンテナンスが容易になるようにする、高度なロケーター生成メカニズムが備えられています。使用するアプリケーションやフレームワークに応じて、最適な結果を得るためにデフォルト設定を変更できます。それぞれのテクノロジーで使用できる任意のプロパティ (整数や倍精度の数値、文字列、項目識別子、列挙値) を、カスタム属性として使用できます。


頻繁には変更されない属性を利用して、適切に定義されたロケーターでは、メンテナンス作業が少なく抑えられます。カスタム属性を使用すると、caption や index などの他の属性を使用するよりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は他のオブジェクトが追加されると変更される可能性があるためです。

カスタム属性 タブのリスト ボックスに一覧表示されているテクノロジー ドメインの場合、任意のプロパティ (*myCustomProperty* を定義する WPFButton など) を取得し、それらのプロパティをカスタム属性として使用することもできます。最適な結果を得るために、テストで利用する要素にカスタム オートメーション ID を追加します。Web アプリケーションでは、操作する要素に `<div myAutomationId= "my unique element name" />` などの属性を追加できます。また、Java SWT では、GUI を実装する開発者が属性 (testAutomationId など) をウィジェットに対して定義することによって、アプリケーション内でそのウィジェットを一意に識別できます。テスト担当者は、その属性をカスタム属性 (この場合は testAutomationId) のリストに追加し、その一意の ID によってコントロールを識別できます。この手法によって、ロケーターの変更に伴うメンテナンス作業を回避することができます。

caption のように、複数のオブジェクトで同じ属性値が共有されている場合、Silk Test Workbench は、複数の利用可能な属性を "and" 操作で結合してロケーターを一意にするよう試み、一致したオブジェクトのリストを単一のオブジェクトになるまで絞り込んでいきます。それができなくなった場合には、索引を付加します。つまり、ロケーターは caption が xyz である *n* 番目のオブジェクトを探すことを意味します。


複数のオブジェクトに同じカスタム属性の値が割り当てられた場合は、そのカスタム属性を呼び出したときにその値を持つすべてのオブジェクトが返されます。たとえば、一意の ID として loginName を 2 つの異なるテキスト フィールドに割り当てた場合は、loginName 属性を呼び出したときに、両方のフィールドが返されます。


1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. テストするアプリケーションのテクノロジー ドメインをクリックします。
たとえば、SAP アプリケーションをテストする場合は、メニュー ツリーで **SAP** をクリックします。


 **注:** Flex または Windows API ベースのクライアント/サーバー (Win32) アプリケーションには、カスタム属性を設定できません。

4. **カスタム属性** テキスト ボックスに、使用する属性を入力します。
カスタム属性が利用可能な場合は、ロケーター生成プログラムは、他の属性の前にそれらの属性を使用します。リストの順番は、ロケーター生成プログラムが使用する属性の優先順位を表しています。指定した属性が選択したオブジェクトに対して利用可能ではなかった場合には、Silk Test Workbench はテストしているアプリケーションのデフォルトの属性を使用します。

複数の属性名を指定する場合にはカンマで区切ります。

 **注:** Web アプリケーションにカスタム属性を含めるためには、HTML タグとして追加します。たとえば、bcauid という属性を追加するには、`<input type='button' bcauid='abc' value='click me' />` と入力します。

 **注:** Java SWT コントロールにカスタム属性を含めるには、`org.swt.widgets.Widget.setData(String key, Object value)` メソッドを使用します。

 **注:** Swing コントロールにカスタム属性を含めるには、`putClientProperty("propertyName", "propertyValue")` メソッドを使用します。

5. **OK** をクリックします。

記録/再生の対象とする WPF クラスの設定

Silk Test Workbench は、一般的に機能テストに無関係なコントロールは除外します。たとえば、レイアウトを目的として使用されるコントロールは含まれません。しかし、カスタム コントロールが除外されたクラスから派生している場合、除外されたコントロールを記録/再生の対象とするためには、関連する WPF クラスの名前を指定します。

記録や再生の対象にしたい WPF クラスの名前を指定します。たとえば、*MyGrid* というカスタム クラスが WPF Grid クラスから継承された場合、*MyGrid* カスタム クラスのオブジェクトは記録や再生に使用できません。Grid クラスはレイアウト目的のためにのみ存在し、機能テストとは無関係であるため、Grid オブジェクトは記録や再生に使用できません。この結果、Grid オブジェクトはデフォルトでは公開されません。機能テストに無関係なクラスに基づいたカスタム クラスを使用するには、カスタム クラス (この場合は *MyGrid*) を **OPT_WPF_CUSTOM_CLASSES** オプションに追加します。これによって、記録、再生、検索、プロパティの検証など、すべてのサポートされる操作を指定したクラスに対して実行できるようになります。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。
3. **WPF** をクリックします。
4. **カスタム WPF クラス名** グリッドで、記録や再生中に公開するクラスの名前を入力します。
複数のクラス名を指定する場合にはカンマで区切ります。
5. **OK** をクリックします。

記録の出力オプションの設定

記録の出力 オプションを変更して、記録中にコントロールをキャプチャするかどうか、プロパティを記録するかどうか、画面のキャプチャ方法、個別の画面をキャプチャするかどうかを指定します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **出力** の隣にあるプラス記号 (+) をクリックします。 **ビジュアル テスト** サブカテゴリが表示されます。
4. **ビジュアル テスト** をクリックします。
5. オプション パネルの最初の列でオプションの名前を選択し、2 番目の列で新しい値を選択または入力して、これらのオプションを変更します。

オプションを選択すると、その簡単な説明が下のパネルに表示されます。

以下の出力オプションを設定できます。

画面キャプチャ Silk Test Workbench で記録中にテスト アプリケーションの画面をキャプチャする方法を指定します。

テストアプリケーションおよびテストアプリケーション内のすべてのウィンドウをキャプチャする場合は、**アプリケーション** に設定します。

テストアプリケーションのウィンドウのうち、記録時にアクティブなウィンドウのみをキャプチャするには、**アクティブ ウィンドウ** に設定します。

すべてのアプリケーション (テスト アプリケーションを含む) の画面のうち、記録時にデスクトップに表示される画面をキャプチャするには、**デスクトップ** に設定します。

記録時に画面をキャプチャしない場合は、**なし** に設定します。

画面キャプチャ (テストステップ)

画面プレビュー に表示される個別の画面をキャプチャする場合は、**はい** に設定します。これによって、**テスト ステップ** ペインの個々のテストステップと連動して、**画面プレビュー** の個々の画面で発生する操作を表示できるようになります。また、再生時に記録される画面を、ビジュアル テストを最初に記録したときにキャプチャされた画面と比較できます (再生時にもテスト ステップの画面がキャプチャされる場合)。

記録中に個別のテスト ステップをキャプチャしない場合は、**いいえ** に設定します。**画面キャプチャ** 設定が **なし** 以外に設定された場合、ダイアログ ボックスまたは Web ページで発生する操作のグループのメイン コンテキストに対して画面がキャプチャされます。この結果、各テストステップは **画面プレビュー** でグループ画面に関連付けられます。

コントロールのキャプチャ

記録中にコントロールとそのプロパティをキャプチャするには、**はい** に設定します。コントロールは、テスト アプリケーションにアクセスしなくても **画面プレビュー** を使用して選択できます。再生のパフォーマンスを向上させるには、このオプションを **いいえ** に設定します。

Web アプリケーションの記録中は、キャプチャされたコントロールとともに以下の属性が格納されます。

- id
- class
- name
- value
- href
- innerText

属性を追加するには、**ツール > オプション > 記録 > xBrowser** の **カスタム属性** フィールドを使用します。



注: 属性のテキスト値は、**ツール > オプション > 記録 > xBrowser** の **属性値の最大の長さ** フィールドで指定した長さに切り詰められます。

ブラウザコントロールのキャプチャ

Web ブラウザの記録中にコントロールとそのプロパティをキャプチャするかどうかを定義します。デフォルト値は、**いいえ** です。**はい** に設定すると、Web ブラウザにアクセスすることなく、**プレビュー画面** でコントロールとそのプロパティを扱うことができます。**コントロールのキャプチャ** オプションが **いいえ** に設定されていると、Silk Test Workbench はこのオプションを無視します。



警告: このオプションを **はい** に設定すると、テストの再生に時間がかかることがあります。再生のパフォーマンスを向上させるには、このオプションを **いいえ** のままにしてください。



ヒント: これらのオプションを **再生結果: ビジュアル テスト** オプションと組み合わせて使用して、再生結果を、ビジュアル テストを最初に記録したときにキャプチャした画面と比較できます。

6. 省略可能 : **名前を付けて保存** をクリックして、**再生** オプションの設定をまとめて保存します。

7. **OK** をクリックします。

記録の出力：ビジュアルテストオプション

記録の出力：ビジュアルテスト オプションを使用して、ビジュアルテストでコントロールおよび画面を記録する場合の動作を制御します。これらのオプションの使用の詳細については、「データベースの膨張の抑制」を参照してください。

以下の出力オプションを設定できます。

画面キャプチャ Silk Test Workbench で記録中にテスト アプリケーションの画面をキャプチャする方法を指定します。

テスト アプリケーションおよびテスト アプリケーション内のすべてのウィンドウをキャプチャする場合は、**アプリケーション** に設定します。

テスト アプリケーションのウィンドウのうち、記録時にアクティブなウィンドウのみをキャプチャするには、**アクティブ ウィンドウ** に設定します。

すべてのアプリケーション (テスト アプリケーションを含む) の画面のうち、記録時にデスクトップに表示される画面をキャプチャするには、**デスクトップ** に設定します。

記録時に画面をキャプチャしない場合は、**なし** に設定します。

画面キャプチャ (テストステップ) **画面プレビュー** に表示される個別の画面をキャプチャする場合は、**はい** に設定します。これによって、**テストステップ** ペインの個々のテストステップと連動して、**画面プレビュー** の個々の画面で発生する操作を表示できるようになります。また、再生時に記録される画面を、ビジュアルテストを最初に記録したときにキャプチャされた画面と比較できます (再生時にもテストステップの画面がキャプチャされる場合)。

記録中に個別のテスト ステップをキャプチャしない場合は、**いいえ** に設定します。**画面キャプチャ** 設定が **なし** 以外に設定された場合、ダイアログ ボックスまたは Web ページで発生する操作のグループのメイン コンテキストに対して画面がキャプチャされます。この結果、各テスト ステップは **画面プレビュー** でグループ画面に関連付けられます。

コントロールのキャプチャ 記録中にコントロールとそのプロパティをキャプチャするには、**はい** に設定します。コントロールは、テスト アプリケーションにアクセスしなくても **画面プレビュー** を使用して選択できます。再生のパフォーマンスを向上させるには、このオプションを **いいえ** に設定します。

Web アプリケーションの記録中は、キャプチャされたコントロールとともに以下の属性が格納されます。

- id
- class
- name
- value
- href
- innerText

属性を追加するには、**ツール > オプション > 記録 > xBrowser** の **カスタム属性** フィールドを使用します。



注: 属性のテキスト値は、**ツール > オプション > 記録 > xBrowser** の **属性値の最大の長さ** フィールドで指定した長さに切り詰められます。

ブラウザコントロールのキャプチャ Web ブラウザの記録中にコントロールとそのプロパティをキャプチャするかどうかを定義します。デフォルト値は、**いいえ** です。**はい** に設定すると、Web ブラウザにアクセスすることなく、**プレビュー画面** でコントロールとそのプロパティを扱うことができます。**コントロールのキャプチャ** オプションが **いいえ** に設定されていると、Silk Test Workbench はこのオプションを無視します。



警告: このオプションを **はい** に設定すると、テストの再生に時間がかかることがあります。再生のパフォーマンスを向上させるには、このオプションを **いいえ** のままにしてください。

記録のホットキー オプションの設定

ホットキー オプションを変更して、記録の開始および停止に使用するキーの組み合わせ、または記録中に検証を挿入するキーの組み合わせを指定します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **ホットキー** をクリックします。

次のようなホットキーの組み合わせを設定できます。

検証の挿入	記録を一時的に停止するキーの組み合わせを指定し、 テスト ロジック デザイナ ウィザードを表示します。ウィザードが開いたあと、検証するオブジェクトを選択します。
現在のオブジェクトを使用して検証を挿入	記録を一時的に停止するキーの組み合わせを指定し、現在ハイライトされているオブジェクトを検証用を選択します。このキーの組み合わせでは、 テスト ロジック デザイナ ウィザードが開き、選択されたオブジェクト セットが検証対象のコントロールとなります。
記録の開始/停止	記録のオン/オフを切り替えるために使用するキーの組み合わせを指定します。

4. ホットキーの組み合わせを変更するには、以下のステップを実行します。
 - a) 右側のオプション パネルで、変更するホットキー オプションを選択します。
 - b) 選択した操作の横にある省略記号ボタンをクリックします。 **ホットキーの定義** ダイアログ ボックスが開きます。
 - c) 修飾キーを使用するには、該当するチェック ボックスをオンにします。
 - d) **キー** リストから、ホットキーとして使用するキーを選択します。
キーの値は、A から Z、0 から 9、F1 から F12 です。
Silk Test Workbench により、このキー シーケンスがテキスト ボックスに入力されます。
 - e) **OK** をクリックします。
5. **OK** をクリックします。

記録オプションプロファイルの作成

記録環境をすばやく構成できるように、記録オプションのデフォルト値を変更したあと、各オプションへの変更を記録オプション プロファイルに保存できます。プロファイルを作成すると、テスト スクリプトまたはビジュアル テストを実行する前に、保存した記録オプションを適用できます。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **全般**、**ロケータ**、**xBrowser**、**SWT**、**出力**、または **ホットキー** をクリックし、**記録** オプションを変更して、Silk Test Workbench が記録中にアプリケーションの動作を識別してキャプチャする方法を変更します。
4. **名前を付けて保存** をクリックします。 **記録設定** ダイアログ ボックスが開きます。
5. **名前** テキスト ボックスで、記録オプション設定の現在のセットの名前を入力するか、リストから既存のグループを選択します。

既存のグループを選択すると、この名前ですべて以前に保存した設定が現在の設定によって置換されます。

6. **所有者** テキスト ボックスで、これらの設定のユーザーを選択します。


7. **OK** をクリックします。

保存済み記録オプションの適用

記録オプション プロファイルを適用するには、記録オプション プロファイルを事前に作成します。

プロファイルに記録オプションのセットを保存して、それらをあとでまとめて適用します。この機能を使用すると、特定の環境に対してそれぞれ異なる記録オプション セットを維持できます。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。 **記録** オプションが右側のパネルに表示されます。
3. **全般、ロケーター、xBrowser、SWT、出力**、または **ホット キー** をクリックします。
4. **記録設定** リストから、適用する保存済みオプションの名前を選択します。元の設定を復元する場合は、**システム デフォルト** を選択します。

 **注:** 既存のオプション プロファイルを変更する場合は、オプション プロファイルを正しく更新してください。既存のオプション プロファイルを変更するには、使用する値にオプションを設定して **名前を付けて保存** をクリックし、適切なオプション プロファイル名を指定する必要があります。このようにしない場合、選択したオプションはダイアログ ボックスに保存されますが、オプション プロファイルには適用されません。

Silk Test Workbench では、保存済みプロファイルの設定によって、現在の **記録** オプションが更新されます。

5. **OK** をクリックします。

再生オプション

再生オプションは、以下のとおりです。

全般	オブジェクトが呼び出されるタイミング、再生中にオブジェクトをハイライトするかどうか、応答のないアプリケーションを閉じるかどうか、再生モード、 出力 ウィンドウにスクリプトのオートメーション呼び出しを表示するかどうか、アプリケーションがアクティブになるまで待機する時間を指定します。
タイミング	アプリケーションが準備完了状態になるまで待機する時間 (ミリ秒)、ウィンドウのスクリーンショットを取得する前に待機する時間 (ミリ秒)、2 つの列挙をキャプチャする間に経過可能な時間、再生中のキー入力間の遅延、スクリプトの各マウス イベントの前の遅延、再生中にオブジェクトが解決されるまで待機する時間、および応答のない再生操作をキャンセルするまでの時間を指定します。
結果	デフォルトの結果名、結果の番号付け方式、合格条件のパーセンテージ、すべての情報を保存するかどうか、再生中にビジュアル テストのコントロールおよび画面をキャプチャするかどうかを設定します。
ホット キー	ツール バーのボタン、資産のダイアログ ボックス、またはショートカット キーの組み合わせ (Ctrl+S) で資産を保存するときの、デフォルトの保存動作を設定します。動作は、資産の新しいバージョンを作成する、または資産の現在のバージョンを置き換える、のいずれかから選択します。
xBrowser	同期から除外する URL、同期モード、および Web アプリケーションのタイムアウト値を指定します。
閉じる	ウィンドウやダイアログ ボックスを閉じるボタン、メニュー項目、確認ダイアログ ボックス、キー入力のリストを指定します。

再生ステータスダイアログ	テストの再生時に、 再生ステータス ダイアログ ボックスを表示するかどうか、および、ダイアログ ボックスにスクリーンショットを表示するかどうかを指定します。
Open Agent の設定	エージェントに直接送信されるカスタム設定を行います。サポートに勧められた場合など、非常に特殊なテスト シナリオでのみ設定されるべきです。

全般再生オプションの設定

オブジェクトが呼び出されるタイミング、再生中にオブジェクトをハイライトするかどうか、応答のないアプリケーションを閉じるかどうか、再生モード、**出力** ウィンドウにスクリプトのオートメーション呼び出しを表示するかどうか、アプリケーションがアクティブになるまで待機する時間を指定します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。
3. **全般** をクリックします。

例外がスローされたときにキャッチする この設定の目的は、発生した例外にステップインできるようにすることです。スクリプトの作成とテストを行っている場合は、これを **はい** に設定して問題の場所を確認できるようにします。自動スクリプトを実行している場合は、独自のエラー処理を備える必要があるため、この設定を **いいえ** にします。エラー処理をスクリプトでオンに切り替えた場合は、**続行** をクリックしてそれにアクセスします。この設定を **いいえ** にして独自のエラー処理を備えていない場合は、スクリプトを **終了** することしかできません。

自動 に設定したとき、Silk Test から普通に実行している場合は、この設定は **はい** に設定したかのように動作します。Silk Central Test Manager によって実行されたスクリプトを Silk Test が実行している場合は、**いいえ** に設定したかのように動作します。よって、スクリプトが独自のエラー処理を持ち、オートメーションから実行されている場合は、スクリプトは動作し続けます。

無応答のアプリケーションを閉じる Silk Test エージェントとアプリケーションとの間の通信がエラーになったかタイムアウトになった場合、アプリケーションをシャットダウンします。
インスタンスを複数実行することができないアプリケーションをテストする場合は、**はい** を指定します。

オブジェクトがアクティブであることを保証する 呼び出しの実行前にオブジェクトがアクティブであることを保証します。

画面検証を実行する Rumba 用の画面検証が記録/再生されるかどうかを制御します。

再生中にオブジェクトをハイライトする 再生選択中に現在のオブジェクトをハイライトします。

再生モード 以下のオプションのいずれかを選択します。

デフォルト このモードを使用すると、最も信頼できる結果が得られます。デフォルトでは、各コントロールそれぞれが、マウスやキーボード (低レベル)、あるいは API (高レベル) モードのどちらかを使用します。デフォルトモードを使用すると、各コントロールがコントロールの種類に応じて適切なモードが使用されます。

高レベル このモードを使用すると、API を使用して各コントロールが再生されます。

低レベル このモードを使用すると、マウスとキーボードを使用して各コントロールが再生されます。

出力ウィンドウにオートメーション操作を表示する スクリプトのオートメーション呼び出しを **出力** ウィンドウに表示します。
出力 ウィンドウを開いて結果を確認する必要があります。**表示** > **出力** をクリックします。

ローカル変数ウィンドウにプロパティを表示する この設定では、通常メンバの変数とパラメータとともに、クラスのプロパティを **ローカル変数** ウィンドウに表示するかどうかを制御します。クラスのプロパティには実装コード (計算) が含まれることがあるので、これにエラーが含まれて、その値が問題を示すことがあります。また、プロパティの計算にはメンバの変数の取得よりも長い時間がかかることがあります。デフォルト値は **はい** です。

4. **OK** をクリックします。

xBrowser 同期オプションの設定

Web アプリケーションの同期およびタイムアウトの値を指定します。同期は、すべてのメソッド呼び出しの前後に自動的に実行されます。メソッド呼び出しは、同期条件が満たされるまで開始せず、終了もしません。

1. **ツール** > **オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。
3. **xBrowser** をクリックします。
4. **同期除外リスト** テキスト ボックスに、除外するサービスまたは Web ページの URL 全体あるいは URL の一部を入力します。

AJAX フレームワークやブラウザによっては、サーバーから非同期にデータを取得するために、特殊な HTTP 要求を継続して出し続けるものがあります。これらの要求により、指定した同期タイムアウトの期限が切れるまで同期がハングすることがあります。この状態を回避するには、HTML 同期モードを使用するか、問題が発生する要求の URL を **同期除外リスト** 設定で指定します。

たとえば、クライアントからデータをポーリングすることによってサーバー時間を表示するウィジェットを Web アプリケーションで使用する場合は、このウィジェットのトラフィックが永続的にサーバーに送信されます。このサービスを同期から除外するには、サービス URL を判別し、除外リストに入力します。

たとえば、以下のように入力します。

- http://example.com/syncsample/timeService
- timeService
- UICallbackServiceHandler

複数のエントリをカンマで区切って指定します。



注: アプリケーションで 1 つのサービスのみが使用されている場合、そのサービスでテストを無効にするには、サービス URL を除外リストに追加するのではなく、HTML 同期モードを使用する必要があります。



ヒント: 除外リストには、URL 全体ではなく URL の部分文字列を追加することを、Micro Focus ではお勧めしています。たとえば、http://example.com/syncsample/timeService ではなく、/syncsample を除外リストに追加します。ブラウザは相対 URL のみを Silk Test Workbench に返す場合があるため、URL 全体を指定すると機能しない場合があります。たとえば、ブラウザは /syncsample/timeService だけを返すのに対して、除外リストに http://example.com/syncsample/timeService を追加していると、Silk Test Workbench は返された URL を除外しません。

- 5. 同期モード** リスト ボックスから、Web アプリケーションを準備可能状態にするための同期アルゴリズムを選択します。
同期アルゴリズムは、呼び出しが可能になる状態までの待機時間を設定します。
HTML モードを使用すると、すべての HTML ドキュメントが対話的な状態になることが保証されます。このモードでは、単純な Web ページをテストすることができます。Java Script が含まれる、より複雑なシナリオが使用される場合は、同期関数を手動で記述することが必要になる可能性があります。
AJAX モードを使用すると、同期関数を手動で記述する必要がなくなるため、スクリプト（オブジェクトの表示または非表示を待機したり、特定のプロパティ値を待機するなど）の作成処理が大幅に簡略化されます。また、この自動同期は、スクリプトを手動で適用しないで記録と再生を正常に行うための基礎となります。
- 6. 同期タイムアウト** テキスト ボックスで、オブジェクトが準備可能状態になるまで待機する最大時間をミリ秒で入力します。デフォルト値は、300000 に設定されています。
- 7. iframe サポートの包含リスト** に、テストの対象となる iframe と frame の属性リストを指定します。
リストの各項目で、属性名と対応する値を定義します。項目のどれにも一致しない iframe と frame は、テスト対象から除外されます。ワイルドカードを使用できます。たとえば、"name:*form" という項目を定義すると、<IFRAME name="user-form" src=...> がテスト対象に含まれます。このオプションは、Internet Explorer では無視されます。リストが空の場合は、すべての iframe と frame がテストの対象となります。複数のエントリをカンマで区切って指定します。
- 8. iframe サポートの除外リスト** に、テストから除外する iframe と frame の属性リストを指定します。
リストの各項目で、属性名と対応する値を定義します。項目のどれにも一致しない iframe と frame は、テスト対象に含まれます。ワイルドカードを使用できます。たとえば、"src:*advertising*" という項目を定義すると、<IFRAME src=http://my.domain/advertising-banner.html> がテスト対象から除外されます。このオプションは、Internet Explorer では無視されます。リストが空の場合は、すべての iframe と frame がテストの対象となります。複数のエントリをカンマで区切って指定します。
- 9. OK** をクリックします。

再生のタイミング オプションの設定

再生 オプションを使用して、再生中のタイミングを制御できます。

- 1. ツール > オプション** をクリックします。 **オプション** ダイアログ ボックスが開きます。
- 2. オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。 **再生** オプションが右側のパネルに表示され、**タイミング** サブカテゴリが **再生** の下のアウトライン ビューに表示されます。
- 3. タイミング** をクリックします。
- 4. アプリケーション準備完了タイムアウト** テキスト ボックスで、アプリケーションが準備完了状態になるまで待機する時間をミリ秒で指定します。
指定したタイムアウト時間内にアプリケーションの準備が整わない場合は、Silk Test Workbench によって例外がスローされます。
- 5. キーボード イベント遅延** テキスト ボックスで、再生中のキー入力間の遅延をミリ秒で指定します。テストするアプリケーションに応じて、選択する最適な値は異なります。たとえば、Web アプリケーションをテストする場合、1 ミリ秒に設定すると、ブラウザが極端に遅くなります。ただし、この値を 0 (ゼロ) に設定すると、基本的なアプリケーション テストに失敗する可能性があります。
- 6. マウス イベント遅延** テキスト ボックスで、スクリプトの各マウス イベントの前に使用される遅延をミリ秒で指定します。
この遅延は、マウスを移動したり、ボタンを押したり離したりする操作に影響します。
- 7. オブジェクト有効化タイムアウト** テキスト ボックスで、再生中にオブジェクトが有効になるまで待機する時間をミリ秒で指定します。
- 8. オブジェクト解決タイムアウト** テキスト ボックスで、再生中にオブジェクトが解決されるまで待機する時間をミリ秒で指定します。
- 9. スクリーンショット遅延** の値を指定します。

新しいウィンドウにフォーカスを設定してからウィンドウのスクリーンショットを取得するまで待機する時間をミリ秒単位で入力します。

デフォルト値は、300 ミリ秒に設定されています。

10無応答のアプリケーションのタイムアウト テキスト ボックスで、応答のない再生操作をキャンセルするまでの時間をミリ秒で入力します。

11省略可能：名前を付けて保存 をクリックして、**再生** オプションの設定をまとめて保存します。

12OK をクリックします。

UI オートメーション オプションの設定

Microsoft UI オートメーション (MSUIA) サポートの有効/無効の切り替え、ロケータ生成用のカスタム属性の指定、ロケータから除外する属性の指定を行います。

UI オートメーション サポートを有効にして、ロケータから除外する属性と値を指定することで、UI オートメーション プロバイダー インターフェイスを実装するアプリケーション (JavaFX や QT ベースのアプリケーションなど) のオブジェクトを、より正確に Silk Test Workbench が識別できるようにすることができます。

Silk Test Workbench には、ロケータが記録時に一意となり、メンテナンスが容易になるようにする、高度なロケータ生成メカニズムが備えられています。使用するアプリケーションやフレームワークに応じて、最適な結果を得るためにデフォルト設定を変更できます。それぞれのテクノロジーで使用できる任意のプロパティ (整数や倍精度の数値、文字列、項目識別子、列挙値) を、カスタム属性として使用できます。

また、UI オートメーション プロバイダー インターフェイスを実装するアプリケーションの任意のプロパティをカスタム属性として設定して使用することもできます。最適な結果を得るために、テストで利用する要素にカスタム オートメーション ID を追加します。

1. **ツール > オプション** をクリックします。

2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。

3. **UI オートメーション** をクリックします。

4. 通常の Win32 コントロールの認識の代わりに Microsoft UI オートメーション サポートも有効化する場合は、**Microsoft UI オートメーション サポートの有効化** を **はい** に設定します。



注: Java FX アプリケーションをテストする場合は、UI オートメーション サポートをオンにする必要はありません。Silk Test Workbench では、Java FX アプリケーションに対して、デフォルトでオンになります。



注: UI オートメーション サポートは、標準のテクノロジー ドメイン固有のサポートをオーバーライドします。UI オートメーション サポートが必要なコントロールに対する操作を完了したら、UI オートメーション サポートをオフにして、標準コントロールに対する操作に戻すこともできます。

5. **オプション** メニュー ツリーの **記録** の隣にあるプラス記号 (+) をクリックします。**記録** オプションが右側のパネルに表示されます。

6. **UI オートメーション** をクリックします。

7. UI オートメーション プロバイダー インターフェイスを実装するアプリケーションのカスタム属性を追加する場合は、**カスタム属性** テキスト ボックスに使用する属性を入力します。

カスタム属性を使用すると、caption や index のような他の属性よりも高い信頼性を得ることができます。これは、caption はアプリケーションを他の言語に翻訳した場合に変更され、index は定義済みのウィジェットより前に他のオブジェクトが追加されると変更される可能性があるためです。

8. **ロケータ属性名除外リスト** グリッドで、記録中に無視する属性名を入力します。

たとえば、automationid という名前の属性を持つコントロールが AUT に複数存在するため無視したい場合は、automationid 属性名をリストに追加します。

複数の属性名を指定する場合にはカンマで区切ります。

9. **ロケータ属性値除外リスト** グリッドで、記録中に無視する属性値を入力します。

たとえば、すべての JavaFX コントロールに JavaFX<番号> という属性値があるため、デフォルトで値 JavaFX* がリストに追加されています。

複数の属性名を指定する場合にはカンマで区切ります。

100K をクリックします。

閉じるオプションの設定

ウィンドウやダイアログ ボックスを閉じるためのボタン、メニュー項目、キー入力のリストを指定します。確認ダイアログ ボックスを閉じるためのボタンのリストも指定します。また、ウィンドウのクローズ方法を試行するまでの間隔も指定します。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの 再生 の隣にあるプラス記号 (+) をクリックします。再生 のサブカテゴリが表示されます。
3. 閉じる をクリックします。
4. ウィンドウを閉じるためのボタン テキスト ボックスで、Close、CloseWindows、および Exit で閉じられるウィンドウを閉じるために使用するボタンのリストを指定します。
5. 確認ダイアログに対するボタン テキスト ボックスで、確認ダイアログ ボックス (Close、CloseWindows、および Exit でウィンドウを閉じるときに表示されるダイアログ ボックス) を閉じるために使用するボタンのリストを指定します。
6. ダイアログを閉じるためのキー テキスト ボックスで、ダイアログ ボックス (Close、CloseWindows、および Exit で使用される) を閉じるために使用するキー シーケンスを指定します。
7. ウィンドウを閉じるためのメニュー項目 テキスト ボックスで、Close、CloseWindows、および Exit でウィンドウを閉じるために使用するメニュー項目のリストを指定します。
8. ウィンドウを閉じるためのタイムアウト の値を指定します。
ウィンドウ閉じるための次の方式を試行する前に待機する時間をミリ秒で入力します。最終的に失敗と判断する前に、4 種類の方式が実行されます。つまり、閉じるのに失敗するまでにかかる合計時間は、指定した値の 4 倍の時間になります。
9. OK をクリックします。

再生結果オプションの設定

再生結果 オプションを変更して、デフォルトの結果名、結果の番号付け方式、合格条件のパーセンテージ、すべての情報を保存するかどうかを設定します。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの 再生 の隣にあるプラス記号 (+) をクリックします。再生 のサブカテゴリが表示されます。
3. 結果 をクリックします。結果 オプションが右側のペインに表示されます。
4. 環境に適したオプションを設定します。

デフォルトの結果名 各結果の結果名として使用する名前を入力します。ビジュアル テストまたはテスト スクリプトの名前を結果名として使用する場合は、このフィールドを空のままにします。

結果オプション 以下のリストから選択します。

- **新しい実行番号で結果を保存** : ビジュアル テストまたはテスト スクリプトの再生の結果は、新しいインクリメント実行番号で保存されます。たとえば、新規の結果は実行番号 1 で保存されます。同じビジュアル テストまたはテスト スクリプトの後続の結果は、実行番号 2、3、4 など保存されます。これはデフォルトの選択です。

- **現在の実行番号に結果を追加**：現在の再生の結果は、最新の既存の実行番号の結果に追加されます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。
- **既存の実行番号を新しい結果で上書き**：現在の再生の結果で、最新の既存の実行番号の結果が上書きされます。新しい実行番号は作成されません。このオプションは、結果が存在している場合にのみ表示されます。
- **この結果を保存しない**：ビジュアルテストまたはテストスクリプトを再生したときに結果は作成されません。

結果合格条件 (パーセンテージ)

今後すべての実行の合格基準として、検証の合格の割合を設定します。たとえば、結果の合格条件を 90% に設定した場合は、ビジュアルテストにおける 10 件の検証のうち 9 件以上に合格しないと、再生の結果が合格になりません。

すべての情報を保存

結果に、ビジュアルテストの再生に関するすべての情報を保存するか、要約情報とエラー情報のみを保存するかを決定します。

5. 省略可能：名前を付けて保存 をクリックして、再生 オプションの設定をまとめて保存します。
6. OK をクリックします。

ビジュアルテストの再生結果オプションの設定

再生結果：ビジュアルテスト オプションを使用して、再生中にビジュアルテストでキャプチャするコントロールおよび画面を設定します。これらのオプションの使用の詳細については、「データベースの膨張の抑制」を参照してください。

1. ツール > オプション をクリックします。
2. オプション メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。
3. **結果** の隣にあるプラス記号 (+) をクリックし、**ビジュアルテスト** をクリックします。**ビジュアルテスト** オプションが右側のペインに表示されます。
4. 環境に適したオプションを設定します。

コントロールのキャプチャ

再生中にコントロールとそのプロパティをキャプチャします。

ビジュアルテストを記録したときに **記録の出力** オプションで設定されていた **コントロールのキャプチャ** 設定と同じ設定を使用するには、**記録と同じ** に設定します。

再生中にコントロールとそのプロパティをキャプチャするには、**はい** に設定します。コントロールは **画面プレビュー** を使用して選択できます。



注：記録の **画面キャプチャ** 設定を **なし** に設定した場合は、**コントロールのキャプチャ** を **はい**、再生の **画面キャプチャ** 設定を **なし** 以外の値に設定して、再生時にコントロールをキャプチャする必要があります。

再生のパフォーマンスを向上させるには、**いいえ** に設定します。

画面キャプチャ

Silk Test Workbench で再生中にテストアプリケーションの画面をキャプチャする方法を指定します。


ビジュアルテストを記録したときに各画面に設定されていた **画面キャプチャ** 設定と同じ設定を使用するには、**記録と同じ** に設定します。たとえば、**アプリケーション** と **アクティブウィンドウ** の両方が設定された状態で記録された画面が含まれるビジュアルテストは、ビジュアルテストで各画面に適用されたように、これらの設定を使用して再生されます。

テストアプリケーションおよびテストアプリケーション内のすべてのウィンドウをキャプチャする場合は、**アプリケーション** に設定します。

テスト アプリケーションのウィンドウのうち、再生時にアクティブなウィンドウのみをキャプチャするには、**アクティブ ウィンドウ** に設定します。


すべてのアプリケーション (テスト アプリケーションを含む) の画面のうち、再生時にデスクトップに表示される画面をキャプチャするには、**デスクトップ** に設定します。

再生時に画面のキャプチャを無効にするには、**なし** に設定します。

 **注:** この設定を **なし** に設定すると、**コントロールのキャプチャ** および **画面キャプチャ (テスト ステップ)** 設定に指定した値に関係なく、コントロールまたは画面は **結果** ウィンドウにキャプチャされません。

画面キャプチャ (テスト ステップ) **画面プレビュー** に表示される個別の画面をキャプチャする場合は、**はい** に設定します。これによって、**テスト ステップ** ペインの個々のテスト ステップと連動して、**画面プレビュー** の個々の画面で発生する操作を表示できるようになります。また、再生時に記録される画面を、ビジュアル テストを最初に記録したときにキャプチャされた画面と比較できません (記録時にもテスト ステップの画面がキャプチャされた場合)。

再生中に個別のテスト ステップをキャプチャしない場合は、**いいえ** に設定します。**画面キャプチャ** 設定が **なし** 以外に設定された場合、個別のテスト ステップに関連付けられたグループ画面が **画面プレビュー** に表示されます。

 **ヒント:** これらのオプションを **記録の出力** オプションと組み合わせて使用して、再生結果を、ビジュアル テストを最初に記録したときにキャプチャした画面と比較できます。

5. 省略可能 : **名前を付けて保存** をクリックして、**再生** オプションの設定をまとめて保存します。

6. **OK** をクリックします。

.NET スクリプトの再生結果オプションの設定

再生結果 : .NET スクリプト オプションを使用して、.NET スクリプトの再生時の画面キャプチャを設定できます。

1. **ツール > オプション** をクリックします。

2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。**再生** のサブカテゴリが表示されます。

3. **結果** の隣にあるプラス記号 (+) をクリックし、**.NET スクリプト** をクリックします。**.NET スクリプト** オプションが右側のペインに表示されます。

4. **画面キャプチャ** リストから、再生中にテスト アプリケーションの画面を Silk Test Workbench がキャプチャする方法を選択します。

- 再生時に画面のキャプチャを無効にするには、**なし** に設定します。
- テスト アプリケーションおよびテスト アプリケーション内のすべてのウィンドウをキャプチャする場合は、**アプリケーション** に設定します。
- テスト アプリケーションのウィンドウのうち、再生時にアクティブなウィンドウのみをキャプチャするには、**アクティブ ウィンドウ** に設定します。
- すべてのアプリケーション (テスト アプリケーションを含む) の画面のうち、再生時にデスクトップに表示される画面をキャプチャするには、**デスクトップ** に設定します。

5. 省略可能 : **名前を付けて保存** をクリックして、**再生** オプションの設定をまとめて保存します。

6. **OK** をクリックします。

キーワード駆動テストの再生結果オプションの設定

再生結果 : キーワード駆動テスト オプションを使用して、キーワード駆動テストの再生中に画面をキャプチャを設定できます。

1. **ツール > オプション** をクリックします。

2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。 **再生** のサブカテゴリが表示されます。
3. **結果** の隣にあるプラス記号 (+) をクリックし、 **キーワード駆動テスト** をクリックします。 **キーワード駆動テスト** オプションが右側のペインに表示されます。
4. **画面キャプチャ** リストから、再生中にテスト アプリケーションの画面を Silk Test Workbench がキャプチャする方法を選択します。
 - 再生時に画面のキャプチャを無効にするには、 **なし** に設定します。
 - テスト アプリケーションおよびテスト アプリケーション内のすべてのウィンドウをキャプチャする場合は、 **アプリケーション** に設定します。
 - テスト アプリケーションのウィンドウのうち、再生時にアクティブなウィンドウのみをキャプチャするには、 **アクティブ ウィンドウ** に設定します。
 - すべてのアプリケーション (テスト アプリケーションを含む) の画面のうち、再生時にデスクトップに表示される画面をキャプチャするには、 **デスクトップ** に設定します。
5. 省略可能 : **名前を付けて保存** をクリックして、 **再生** オプションの設定をまとめて保存します。
6. **OK** をクリックします。

再生ホットキーの変更

Silk Test Workbench が最小化されているときに再生を停止するための再生ホット キーを変更します。

[オプション] メニュー ツリーで、 **再生** の隣にあるプラス記号 (+) をクリックし、 **ホット キー** をクリックします。 **再生の停止** を選択して、キーの指定を編集します。 デフォルトでは、再生を停止するホット キーは Alt+F12 です。

再生オプションプロファイルの作成

プロファイルに再生オプションのセットを保存して、それらをあとでまとめて適用できます。 この機能を使用すると、特定の環境に対してそれぞれ異なる再生オプション セットを維持できます。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。 **再生** のサブカテゴリが表示されます。
3. **タイミング**、 **結果**、 **ビジュアルテスト**、または **ホット キー** をクリックし、 **再生** オプションを変更して、Silk Test Workbench が再生中にアプリケーションの動作を識別してキャプチャする方法を変更します。
4. **名前を付けて保存** をクリックします。 **再生設定** ダイアログ ボックスが開きます。
5. **名前** テキスト ボックスで、再生オプション設定の現在のセットの名前を入力するか、ドロップダウン リストから既存のグループを選択します。 既存のグループを選択すると、この名前ですべて以前に保存した設定が現在の設定によって置換されます。
6. **所有者** テキスト ボックスで、これらの設定のユーザーを選択します。
7. **OK** をクリックします。

保存済み再生オプションの適用

再生記録オプション プロファイルを適用するには、再生オプション プロファイルを事前に作成する必要があります。

テスト基準に適合するように、デフォルトの再生オプションを復元するか、またはテスト再生用に作成した新規プロファイルを適用します。

1. **ツール > オプション** をクリックします。
2. **オプション** メニュー ツリーの **再生** の隣にあるプラス記号 (+) をクリックします。 **再生** のサブカテゴリが表示されます。

3. **タイミング**、**結果**、**ビジュアルテスト**、または **ホットキー** をクリックします。
4. パネル最下部の **再生設定** リストから、適用する保存済みオプションの名前を選択します。Silk Test Workbench の元の設定を復元する場合は、**システムデフォルト** を選択します。



注: 既存のオプションプロファイルを変更する場合は、オプションプロファイルを正しく更新してください。既存のオプションプロファイルを変更するには、使用する値にオプションを設定して **名前を付けて保存** をクリックし、適切なオプションプロファイル名を指定する必要があります。このようにしない場合、選択したオプションはダイアログボックスに保存されますが、オプションプロファイルには適用されません。

Silk Test Workbench では、保存済みプロファイルの設定によって、現在の **再生** オプションが更新されます。

5. **OK** をクリックして、新しいオプション設定をアクティブにします。

スクリプト作成オプション

スクリプト作成 オプションによって、フォントの種類やサイズなどのスクリプトのレイアウトを制御します。

1. **ツール > オプション** をクリックします。
2. **オプション** ツリーで **スクリプト作成** をクリックします。スクリプト作成オプションが右側のパネルに表示されます。
3. **フォント** テキストボックスで、省略記号ボタンをクリックします。**フォント** ダイアログボックスが開きます。
4. スクリプトで使用するフォントの種類、スタイル、サイズを選択し、**OK** をクリックします。
5. **OK** をクリックします。

詳細オプションの設定

詳細オプションでは、フォールバックサポートを有効化したり、ロケーター属性名の大文字小文字を区別するかどうか、などを設定できます。**詳細設定 > アクティブデータ** オプションを変更することにより、**アクティブデータ資産の設定** ウィンドウを使用してテキストベース (.csv または .txt) のアクティブデータファイルを編集する際に、そのファイルに Silk Test Workbench からデータを書き込む方法を制御することができます。

1. **ツール > オプション** をクリックします。**オプション** ダイアログボックスが表示されます。
2. **詳細設定** タブをクリックします。**詳細オプション** ページが表示されます。
3. 埋め込み Chrome アプリケーションをテストするには、**埋め込み Chrome サポートの有効化** フィールドに実行可能ファイルとポートを値ペア形式で指定します。
たとえば、myApp.exe=9222 のように指定します。
複数の埋め込み Chrome アプリケーションを指定する場合は、複数の値ペアをカンマで区切って指定します。
4. デフォルトのブラウザーサポートではテストできないハイブリッドモバイルアプリケーションに対するフォールバックサポートを使用するには、**Android と iOS 上で Web ビューのフォールバックサポート** を有効化します。
5. 標準の Win32 コントロールの解決に加えて、Microsoft ユーザー補助を有効にするには、**Microsoft ユーザー補助** を有効化にします。
6. テキストをキャプチャする前にウィンドウからフォーカスを外すには、**テキストのキャプチャ時にフォーカスを外す** を有効化します。
テキストのキャプチャは、次のメソッドによる記録および再生中に実行されます。

- TextClick
- TextCapture
- TextExists
- TextRect

7. ロケータ属性名の大文字と小文字が区別されるように設定するには、**ロケータ属性の大文字小文字を区別する**を有効化します。モバイル Web アプリケーションのロケータ属性の名前は、常に大文字と小文字の区別はされません。つまり、モバイル Web アプリケーションの記録や再生時に、このオプションは無視されます。

8. Mozilla Firefox で外部リンクを新しいウィンドウではなく新しいタブで開く場合は、**Firefox のシングルウィンドウモードを使用する**をオンにします。



注: このオプションは、Mozilla Firefox 52 以降を使用している場合にのみ動作します。

9. オプション メニュー ツリーの **詳細設定** の隣にあるプラス記号 (+) をクリックし、**アクティブ データ** をクリックします。 **アクティブ データ** のオプションが右側のパネルに表示されます。

10環境に適したオプションを設定します。

フィールドを常に引用符で囲む

新しいアクティブ データ ファイルの各セル内のデータを引用符で囲むには、**はい** を選択します。このオプションは新しいアクティブ データ ファイルにのみ関連します。このオプションを **はい** に設定しても、既存のファイルには影響しません。

アクティブ データ ファイルから使用する予定のデータに引用符が必要な場合に、このオプションを **はい** に設定します。



注: **はい** に設定しても、編集中は、新しいアクティブ データ ファイルのデータに引用符は表示されません。ファイルの元のアプリケーションを使ってアクティブ データ ファイルを編集する場合に、引用符が表示されます。

空の後続フィールドを追加する

各データ項目に対して文字の後ろに後続スペースを入力する場合は、**はい** に設定します。たとえば、フィールドに必要なデータが「data」の場合にこのオプションを **はい** に設定すると、後続のスペースはデータの一部として含まれるため、アクティブ データ ファイルのフィールドには「data」と書き込まれます。**いいえ** を選択すると、後続スペースを除いた文字のみが含まれます。たとえば、**いいえ** に設定した場合に「data」がフィールドに書き込まれると、アクティブ データ ファイルには「data」として保存されます。

アクティブ データ テストで使用されるデータで、アプリケーションを正しくテストするために後続スペースが必要な場合、このオプションを **はい** に設定してください。

11OK をクリックします。

アクティブ データ詳細設定オプション


詳細設定 **アクティブ データ** オプションを変更することにより、**アクティブ データ資産の設定** ウィンドウを使用してテキストベース (.csv または .txt) のアクティブ データ ファイルを編集する際に、そのファイルに Silk Test Workbench からデータを書き込む方法を制御することができます。

アクティブ データに対して以下のオプションを設定できます。

フィールドを常に引用符で囲む

新しいアクティブ データ ファイルの各セル内のデータを引用符で囲むには、**はい** を選択します。このオプションは新しいアクティブ データ ファイルにのみ関連します。このオプションを **はい** に設定しても、既存のファイルには影響しません。

アクティブ データ ファイルから使用する予定のデータに引用符が必要な場合に、このオプションを **はい** に設定します。

 **注:** **はい** に設定しても、編集中は、新しいアクティブ データ ファイルのデータに引用符は表示されません。ファイルの元のアプリケーションを使ってアクティブ データ ファイルを編集する場合に、引用符が表示されます。

空の後続フィールドを追加する

各データ項目に対して文字の後ろに後続スペースを入力する場合は、**はい** に設定します。たとえば、フィールドに必要なデータが「data」の場合にこのオプションを **はい** に設定すると、後続のスペースはデータの一部として含まれるため、アクティブ データ ファイルのフィールドには「data」と書き込まれます。**いいえ** を選択すると、後続スペースを除いた文字のみが含まれます。たとえば、**いいえ** に設定した場合に「data」がフィールドに書き込まれると、アクティブ データ ファイルには「data」として保存されます。

アクティブ データ テストで使用されるデータで、アプリケーションを正しくテストするために後続スペースが必要な場合、このオプションを **はい** に設定してください。

Silk Test Workbench の管理

Silk Test Workbench データベース、プロジェクト、およびユーザーの管理手順について説明します。

Silk Test Workbench の実行に必要な管理者権限


Silk Test Workbench のインストールまたは実行には、次の権限が必要です。


- Silk Test Workbench のインストールには、ローカル管理者権限が必要です。
- Windows サーバー上への Silk Test Workbench のインストールには、ドメイン レベルの管理者権限が必要です。
- Silk Test Workbench を実行するには、次のフォルダとそのすべてのサブフォルダに対するフルアクセス権が必要です。
 - C:\ProgramData\Silk\SilkTest
 - %APPDATA%\Roaming\Silk\SilkTest
 - %APPDATA%\Local\Silk\SilkTest
 - %TEMP%

Silk Test Workbench データベースの構成


Silk Test Workbench は、資産や設定を格納するためにデータベースを必要とします。デフォルトの Silk Test Workbench インストールでは、単一ユーザーによる運用用に Microsoft Access データベースが含まれています。本格的なエンタープライズでの運用には、Silk Test Workbench は Microsoft SQL Server および Oracle データベースをサポートします。Microsoft SQL Server または Oracle データベースを Silk Test Workbench で使用するには、中央サーバー マシン上にデータベースを作成して構成し、各クライアント マシン上で中央データベースを対象としたデータ ソース名 (DSN) を作成します。同じプロジェクトに複数のユーザーが関与する長期間にわたる自動化プロジェクトには、Microsoft SQL Server を使用することを Micro Focus は推奨しています。

サポートするデータベースについての情報は、『[リリース ノート](#)』を参照してください。

 **注:** Silk Test Workbench の新しいバージョンをインストールしたときにデータベースが更新された場合、このデータベースを使用するすべての Silk Test Workbench ユーザーが、Silk Test Workbench の新しいバージョンにアップグレードする必要があります。

 **注:** Silk Test では、64 ビットの DSN は使用できません。64 ビット マシンのデフォルトの DSN と ODBC ドライバは、C:\Windows\SysWow64 にある WOW64 ツールを使用して確認できます。

SQL Server データベースの構成

 **重要:** データベースの設定と構成は、データベース管理者またはデータベースの管理に関する一般的な知識を持つ方が行うことをお勧めします。

Silk Test Workbench は、Silk Test Workbench で使用するために構成されたデータベースにテスト資産を格納してアクセスします。このセクションでは、以下の SQL Server データベースを Silk Test Workbench データベースとして使用する場合の構成方法について説明します。

新しい機能、サポートするプラットフォーム、テスト済みのバージョンについての情報は、『[リリース ノート](#)』を参照してください。

SQL Server の要件

Silk Test Workbench で使用するよう SQL Server データベースを構成する場合は、以下の要件があります。

- Silk Test Workbench を実行するすべてのコンピュータは、SQL Server データベースが常駐するコンピュータにアクセスでき、新しいデータ ソースが構成されている必要があります。
- Silk Test Workbench は、データベースにアクセスするすべてのコンピュータにインストールされている必要があります。
- Silk Test Workbench データベース接続ファイルが、SQL Server データベースを指すように構成されている必要があります。
- SQL Server データベースとの接続に使用するユーザーは、データベースの VIEW_SERVER_STATE 権限を持つ必要があります。



注: Silk Test では、64 ビットの DSN は使用できません。64 ビット マシンのデフォルトの DSN と ODBC ドライバは、C:\Windows\SysWow64 にある WOW64 ツールを使用して確認できます。

Silk Test Workbench データベースの保守の詳細については、*Silk Test Workbench* のヘルプを参照してください。

SQL Server データベースの新規作成

このセクションでは、SQL Server Management Studio を使用して SQL Server データベースを作成する方法について説明します。以下の手順以外に、Silk Test Workbench データベース メンテナンス ユーティリティを使用して、データベースに Silk Test Workbench テーブルを入力する必要もあります。

1. SQL Server Management Studio のオブジェクト エクスプローラで、**データベース** フォルダを右クリックして **新しいデータベース** を選択します。 **新しいデータベース** ダイアログ ボックスが開きます。
2. **データベース名** テキスト ボックスにデータベースの名前を入力します。
3. **OK** をクリックします。

SQL Server 管理ユーザーの新規作成

これ以降の設定手順を実行するには、システム管理者権限を持つ SQL Server ユーザーが必要です。

このユーザーには次のアクセス許可が必要です。

- sysadmin または、db_ddladmin と db_owner の両方
- db_datareader
- db_datawriter

これらのアクセス許可は、**CREATE INDEX (Transact-SQL)** 文を使用してインデックスを作成するために必要になります。Silk Test Workbench データベースが作成された後は、データベースに接続するすべてのユーザーに対して、db_datareader と db_datawriter アクセス許可だけが必要になります。


1. システム管理者権限を持つユーザーを新規作成するには、以下のステップを実行します。
 - a) SQL Server Management Studio のオブジェクト エクスプローラで、**セキュリティ** フォルダに移動して、展開します。
 - b) **ログイン** フォルダを右クリックして、**新しいログイン** をクリックします。 **ログイン - 新規作成** ダイアログ ボックスが開きます。
 - c) **全般** ページを選択し、**ログイン名** テキスト ボックスにユーザー名を入力します。
 - d) [SQL Server 認証] を選択し、パスワードを入力します。
 - e) **既定のデータベース** リストから、デフォルト データベースを選択します。
 - f) **サーバー ロール** ページを選択し、**サーバー ロール** リストで **sysadmin** のチェック ボックスをオンにします。
 - g) **ユーザー マッピング** ページを選択します。
 - h) **マップ** 列で、新しいログインでアクセスできるデータベースのチェック ボックスをオンにします。デフォルトで、**ユーザー** 列にログイン名が表示されています。この値はそのままにします。

- i) データベース ロールのメンバシップ リストで、**db_owner** のチェック ボックスをオンにします。
 - j) **OK** をクリックします。
2. 新しいユーザーのスキーマを新規作成するには、以下のステップを実行します。
 - a) **オブジェクト エクスプローラ** で、スキーマを作成するデータベースに移動します。
 - b) データベース オブジェクト ツリーを展開して、**スキーマ** フォルダを表示します。
スキーマ フォルダは、データベースの **セキュリティ** フォルダの子です。
たとえば、スキーマをマスタ データベースに追加する場合は、**データベース > システム データベース > マスタ > セキュリティ** フォルダを展開して **スキーマ** フォルダを表示します。
 - c) **スキーマ** フォルダを右クリックして、**新しいスキーマ** を選択します。
 - d) **スキーマ名** テキスト ボックスに、新しいスキーマの名前を入力します。
新しいスキーマ名は、以前にシステム管理者権限を使用して作成したユーザー名と一致する必要があります。
 - e) スキーマの所有者として新しいユーザーにシステム管理者権限を割り当てます。
 - f) **OK** をクリックします。
 3. データベースにスキーマを追加するには、次の操作を実行します。
 - a) データベース オブジェクト ツリーを展開して、**ユーザー** フォルダを表示します。
ユーザー フォルダは、データベースの **セキュリティ** フォルダの子です。
たとえば、*master* データベースにスキーマを追加する場合は、**データベース > システム データベース > master > セキュリティ > ユーザー** をクリックします。
 - b) 作成した管理ユーザーをダブルクリックします。**データベース ユーザー** ダイアログ ボックスが表示されます。
 - c) **所有されているスキーマ** ページを選択し、作成したスキーマが新しいデータベースに割り当てられるようにチェックします。


SQL Server におけるユーザーの設定


以下の手順は、SQL Server データベースに接続する必要がある各ユーザーに対し、データベース サーバー コンピュータで実行してください。

1. SQL Server Management Studio のオブジェクト エクスプローラで、**セキュリティ** フォルダに移動して、展開します。**ログイン** フォルダを右クリックして、**新しいログイン** をクリックします。**ログイン-新規作成** ダイアログ ボックスが開きます。
2. **全般** ページを選択し、**ログイン名** テキスト ボックスにデータベースの名前を入力します。
3. **Windows 認証** または **SQL Server 認証** を選択します。
4. **既定のデータベース** リストから、デフォルト データベースを選択します。
5. **ユーザー マッピング** ページを選択します。
6. **マップ** 列で、ユーザーのログインでアクセスできるデータベースのチェック ボックスをオンにします。
デフォルトで、**ユーザー** 列にログイン名が表示されています。この値はそのままにします。
7. **既定のスキーマ** 列に、既定のスキーマを入力します。
Silk Test Workbench で既定のスキーマを使用するには、既定のスキーマが、以前にシステム管理者権限を使用して作成したユーザーのスキーマと一致する必要があります。
8. **データベース ロールのメンバシップ** リストで、デフォルト オプション **public** を選択したままにしておきます。
9. **db_datareader** および **db_datawriter** チェック ボックスをオンにします。
10. **OK** をクリックします。

 **重要:** 複数のユーザーが SQL Server Express を使用するには、各 SQL Server Express インストールでリモート接続が有効になっている必要があります。SQL Server Express でリモート接続を有効にする方法については、Microsoft のサポート技術情報『*How to enable remote connections on SQL Server*』を参照してください。


SQL Server Silk Test Workbench データベースの準備

 **注:** SQL Server 認証モードを SQL Server と Windows (混合モード) に設定して、データベース メンテナンス ユーティリティを使用して SQL Server データベースに接続できるようにする必要があります。この設定は、データベース メンテナンス タスクを実行したあと変更できます。

 **注:** SQL Server データベースが空の場合は、このトピックで説明しているタスクのみ実行できます。

Silk Test Workbench で使用する新しい SQL Server データベースを準備する必要があります。このセクションでは、データベース メンテナンス ユーティリティを使用してデータベースに Silk Test Workbench テーブルを入力する方法について説明します。


1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。 **データベース メンテナンス** ユーティリティが起動されます。
2. **ファイル > 新規作成 > SQL Server** をクリックします。 **SQL Server データ ソースの接続** ダイアログ ボックスが開きます。
3. 新しい SQL Server データ ソースの名前を入力するか、**参照** をクリックして **ODBC データ ソースの選択** ダイアログ ボックスからデータ ソース名を選択します。
4. 該当するテキスト ボックスにデータベースの所有者の名前と、SQL Server のユーザー ID およびパスワードを入力し、**作成** をクリックします。データベースの所有者は、db_owner ロールを持つユーザーです。

 **注:** 各 DSN 名に関連付けることができるスキーマは 1 つだけです (ODBC 接続)。選択した DSN 名がすでにスキーマに関連付けられている場合は、**置換** をクリックして DSN 名に関連付けを現在のスキーマから新しいスキーマに変更するか (ユーザーは、古いスキーマの Silk Test Workbench テーブルにはアクセスできなくなります)、**エイリアス** をクリックして DSN 名のエイリアスを作成します。

これで、Silk Test Workbench を起動してログオンする準備が整いました。

Oracle データベースの構成

このセクションでは、Silk Test Workbench で使用する Oracle データベースを設定する方法について説明します。

 **注:** 以下の手順は、その実行者が Oracle データベース管理システムに精通していることを前提としています。データベースの設定と構成は、データベース管理者またはデータベースの管理に関する一般的な知識を持つ方が行うことをお勧めします。

各クライアント コンピュータでは、データベースに接続するために ODBC データ ソースが必要となります。この結果、Silk Test Workbench 用の Oracle データベースに接続するには、以下の条件を満たす必要があります。

- Silk Test Workbench を実行するすべてのコンピュータは、データベースが常駐するコンピュータにアクセスでき、新しいデータ ソースが構成されている必要があります。
- Silk Test Workbench は、データベースにアクセスするすべてのコンピュータにインストールされている必要があります。
- Silk Test Workbench を実行するすべてのコンピュータに、適切なデータベース クライアント接続ソフトウェアがインストールされている必要があります。
- データベース接続は、Silk Test Workbench で使用するために構成する必要があります。

認証方式の選択

Silk Test Workbench では、Oracle OS 認証 (Windows NT) と Oracle データベース認証のいずれかを使用することができます。Oracle OS 認証を使用する場合、追加のインストール手順が必要になります。

Oracle OS 認証

このトピックでは、Oracle OS 認証を使用する場合に、ユーザーを作成して接続権限を付与する方法について説明します。これらのステップを実行する前に、以下の点を考慮してください。


- ORA_DBA グループのメンバであり、かつ、SYSDBA 権限がある必要があります。
- ユーザーを作成する際は、ユーザーが DOMAIN コントローラによって認証されていることを Oracle が認識できるようにする必要があります。
- ユーザーの作成には SQL*Plus または SQL*Plus Worksheet を使用します。SQL*Plus を使い慣れていない場合は、Oracle のマニュアルを参照してください。
- ユーザーは Windows NT ドメイン内に作成し、ORA_DBA グループに追加する必要があります。Windows ドメインの使用の詳細については、Microsoft Windows のマニュアルを参照してください。
- Oracle で Silk Test Workbench のユーザーを設定する場合、リモートで認証されるユーザーに加え、実際のテーブルを格納するためのスキーマを作成します。このスキーマはドメインで認証されず、パスワードで認証されます。

1. SQL*Plus または SQL*Plus Worksheet を起動します。
2. SYSTEM としてログオンし、SYSDBA として接続します。
3. 各ユーザーを "DOMAIN¥USERNAME" IDENTIFIED EXTERNALLY として作成し、接続権限およびシステム権限を付与するコマンドを、以下の例に示すように入力します。

各ユーザーは "DOMAIN¥USERNAME" のように入力します。DOMAIN¥USERNAME は大文字で入力し、二重引用符で囲みます。DOMAIN¥USERNAME は各ユーザーのドメインとユーザー名です。

例：

```
CREATE USER "NT-DOMAIN¥JEFF" IDENTIFIED EXTERNALLY;  
GRANT SELECT ANY SEQUENCE TO "NT-DOMAIN¥JEFF";  
GRANT UNLIMITED TABLESPACE TO "NT-DOMAIN¥JEFF";  
GRANT "CONNECT" TO "NT-DOMAIN¥JEFF";  
GRANT "RESOURCE" TO "NT-DOMAIN¥JEFF";  
GRANT "SELECT_CATALOG_ROLE" TO "NT-DOMAIN¥JEFF";
```

 **重要:** Microsoft のオペレーティング システムでは、ユーザーを照会すると DOMAIN ¥USERNAME が返されます。この値がデータベースの定義と一致しない場合は、Oracle OS 認証で ORA-1017 のエラーが発生します。

4. SQL*Plus を使用してデータベースに接続することにより、Oracle OS 認証が正しく設定されているかどうかをテストします。
 - a) ユーザー名には「/」と入力します。
 - b) [パスワード] テキスト ボックスは空のままにし、ホスト文字列を正しく入力します。
 - c) **OK** をクリックします。ログオンできた場合は、Oracle OS 認証が正しく機能していることを意味します。

Oracle データベース認証

このトピックでは、Oracle データベース認証を使用する場合に、ユーザーを作成して接続権限を付与する方法について説明します。これらのステップを実行する前に、以下の点を考慮してください。

- ORA_DBA グループのメンバであり、かつ、SYSDBA 権限がある必要があります。
- ユーザーの作成には SQL*Plus または SQL*Plus Worksheet を使用します。SQL*Plus を使い慣れていない場合は、Oracle のマニュアルを参照してください。
- Oracle で Silk Test Workbench のユーザーを設定する場合、リモートで認証されるユーザーに加え、実際のテーブルを格納するためのスキーマを作成します。このスキーマはドメインで認証されず、パスワードで認証されます。

1. SQL*Plus または SQL*Plus Worksheet を起動します。
2. SYSTEM としてログオンし、SYSDBA として接続します。
3. 各ユーザーを "USERNAME" IDENTIFIED BY "ORACLEPASSWORD" として作成し、接続権限およびシステム権限を付与するコマンドを、以下の例に示すように入力します。

各ユーザーは "USERNAME" のように入力します。USERNAME は大文字で入力し、二重引用符で囲みます。DOMAIN¥USERNAME は各ユーザーのユーザー名です。

例：

```
CREATE USER "JEFF" IDENTIFIED BY "ORACLEPASSWORD";
GRANT SELECT ANY SEQUENCE TO "JEFF";
GRANT UNLIMITED TABLESPACE TO "JEFF";
GRANT "CONNECT" TO "JEFF";
GRANT "RESOURCE" TO "JEFF";
GRANT "SELECT_CATALOG_ROLE" TO "JEFF";
```


Oracle データベースの新規作成

次の文字セットを使用する新しい Oracle データベースを作成します。

- UTF8：Oracle 12 より前のバージョンの Oracle
- AL32UTF8：Oracle 12 以降


詳細については、Oracle の製品マニュアルを参照してください。

Oracle OS 認証の設定

 **注：** この手順は、Oracle OS 認証を使用する場合にのみ必要です。Oracle データベース認証を使用する場合は、Oracle クライアントをセットアップする必要があります。


1. データベースがあるサーバーで、Oracle をインストールしたディレクトリにある Oracle 初期化ファイルを探します。
2. Oracle 初期化ファイルで、以下のパラメータの値を設定します。

- remote_login_passwordfile = none
- remote_os_authent = true
- os_authent_prefix = ""

 **注：** パラメータと値のうち存在しないものがある場合は、追加する必要があります。Oracle 11g では、remote_os_authent および os_authent_prefix パラメータはデフォルトで設定されています。

3. Oracle をインストールしたディレクトリで、SQLNET.ORA ファイルを探します。
このファイルは、Oracle のメイン インストール ディレクトリの下に ORACLE_HOME¥Network ¥Admin ディレクトリにあります。ORACLE_HOME は、インストール時に Oracle ホームに割り当てた名前です。
4. SQLNET.ORA を開いて、以下のようにパラメータの値を設定します。

```
sqlnet.authentication_services = (NTS)
```

 **注：** このパラメータの行頭が # でコメントアウトされていないことを確認します。このパラメータは、追加する必要がある場合があります。

5. **スタート** メニューから、**ファイル名を指定して実行** ダイアログ ボックスにアクセスします。
6. **名前** テキスト ボックスに「regedit」と入力して **OK** をクリックします。**レジストリ エディタ** ダイアログ ボックスが開きます。
7. HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥HOMEn (n は Oracle インストールに関連付けられた番号) パスで、HOMEn を右クリックして、**新規 > 文字列値** を選択します。レジストリ エディタの右側のウィンドウに、新しい文字列の値が表示されます。
8. デフォルト名の新しい値 #1 を OSAUTH_PREFIX_DOMAIN に置き換えます。
9. 作成した文字列の値をダブルクリックします。**文字列の編集** ダイアログ ボックスが開きます。
- 10 **値のデータ** テキスト ボックスに「TRUE」と入力して **OK** をクリックします。

Oracle クライアントの設定


接続元として使用するクライアント コンピュータで、以下を実行します。

1. Oracle をインストールしたディレクトリで、SQLNET.ORA ファイルを探します。

このファイルは、Oracle のメイン インストール ディレクトリの下の ORACLE_HOME¥Network ¥Admin ディレクトリにあります。ORACLE_HOME は、インストール時に Oracle ホームに割り当てた名前です。

2. SQLNET.ORA を開いて、以下のようにパラメータの値を設定します。

```
sqlnet.authentication_services = (NTS)
```

 **注:** このパラメータの行頭が # でコメントアウトされていないことを確認します。このパラメータは、追加する必要がある場合があります。

3. Oracle をインストールしたディレクトリで、TNSNAMES.ORA ファイルを開きます。


このファイルは ORACLE_HOME¥Network¥Admin ディレクトリにあります。

4. サーバーのデータベース エントリをクライアント コンピュータ上の TNSNAMES.ORA ファイルにコピーします。

例:

```
ORACLE_SID.DOMAIN.COM=  
(DESCRIPTION=  
(ADDRESS_LIST=  
(ADDRESS=(PROTOCOL=TCP)(HOST=SERVERNAME)(PORT=1521))  
)  
(CONNECT_DATA=  
(SERVICE_NAME=ORACLE_SID)  
)  
)
```

クライアント接続用の Oracle データベースの準備

 **注:** この手順は、Oracle OS 認証および Oracle クライアントを使用して Silk Test Workbench データベースを設定する場合にのみ必要です。Oracle データベース認証を使用する場合は、ユーザーの設定を開始することができます。

1. ドメイン ユーザー マネージャがインストールされているコンピュータで、Windows の **コントロールパネル** にアクセスし、**管理ツール > コンピュータの管理** を選択します。

コンピュータの管理 ユーティリティが開きます。

2. ツリー ビューで、**ローカル ユーザーとグループ** をクリックします。

3. **グループ** をクリックします。

Oracle がインストールされている場合、ORA_DBA という名前のグループが表示されます。表示されない場合は、追加する必要があります。

次に、ユーザーに対してサーバー アクセスを作成します。

ORA_DBA グループの作成

1. **コンピュータの管理** ユーティリティのツリー ビューで、**グループ** を右クリックして **新しいグループ** を選択します。 **新しいグループ** ダイアログ ボックスが開きます。
2. **グループ名** テキスト ボックスに「ORA_DBA」と入力します。
3. **作成** をクリックします。
4. **閉じる** をクリックします。

ユーザーに対するサーバー アクセスの作成

ユーザー アカウントを個別に作成するには、以下の手順に従います。

1. **コンピュータの管理** ユーティリティのツリー ビューで、**ローカル ユーザーとグループ** をクリックします。
2. **グループ** をクリックします。
3. 右側のペインで、**ORA_DBA** をダブルクリックします。 **ORA_DBA のプロパティ** ダイアログ ボックスが開きます。
4. **追加** をクリックします。 **ユーザー、コンピュータ、またはグループの選択** ダイアログ ボックスが開きます。
5. **詳細設定** をクリックし、**共通クエリ** タブを使用してユーザーを検索し、**今すぐ検索** をクリックして、リストからユーザーを選択します。

または、**選択するオブジェクト名を入力してください** テキスト ボックスをクリックし、domain name ¥user ID を入力します。

domain name はユーザーが所属するドメインの名前であり、user ID はアクセス権を付与する対象ユーザーのネットワーク ID です。

6. **OK** をクリックしてユーザー情報を保存し、**ORA_DBA のプロパティ** ダイアログ ボックスに戻ります。ステップ 4 から 6 を繰り返して、その他のユーザー アカウントを作成します。
7. **Users のプロパティ** ダイアログ ボックスで **OK** をクリックし、**コンピュータの管理** ユーティリティに戻ります。

Oracle における Silk Test Workbench のユーザーの設定

このセクションでは、ユーザーを作成してユーザーの接続権限を付与する方法について説明します。この手順は Oracle OS 認証または Oracle データベース認証のどちらを使用する場合にも必要です。手順は認証タイプによって異なります。

Oracle OS 認証

ユーザーを作成する際は、ユーザーが DOMAIN コントローラによって認証されていることを Oracle が認識できるようにする必要があります。



重要:

Oracle で Silk Test Workbench のユーザーを設定する場合、リモートで認証されるユーザーに加え、実際のテーブルを格納するためのスキーマも作成する必要があります。このスキーマはドメインで認証されず、パスワードで認証されます。詳細については、「[パスワード認証スキーマの作成](#)」を参照してください。

ユーザーは Windows NT ドメイン内に作成する必要があります。Windows ドメインの使用の詳細については、Microsoft Windows のマニュアルを参照してください。

ユーザーは ORA_DBA グループに追加する必要があります。ユーザーの作成には SQL*Plus または SQL*Plus Worksheet を使用します。SQL*Plus を使い慣れていない場合は、Oracle のマニュアルを参照してください。

以下の手順を実行するユーザーは、ORA_DBA グループのメンバーであり、SYSDBA 権限を持っている必要があります。Oracle データベースにアクセスできるユーザーを指定するには、以下を実行します。

1. SQL*Plus または SQL*Plus Worksheet を起動します。
2. SYSTEM としてログオンし、SYSDBA として接続します。
3. 各ユーザーを作成するコマンドを "DOMAIN¥USERNAME" IDENTIFIED EXTERNALLY として入力し、各ユーザーの接続権限を "DOMAIN¥USERNAME" として付与します。DOMAIN¥USERNAME は大

文字で入力し、二重引用符で囲みます。DOMAIN¥USERNAME は各ユーザーのドメインとユーザー名です。例：

```
SQL> create user "NT-DOMAIN¥JEFF" IDENTIFIED EXTERNALLY;  
SQL> grant connect, resource to "NT-DOMAIN¥JEFF";
```


 **注:** SELECT_ANY_SEQUENCE と SELECT_CATALOG_ROLE もユーザーに付与する必要があります。例：

```
SQL> grant select any sequence to "JEFF";  
SQL> grant select_catalog_role to "JEFF";
```

Microsoft のオペレーティング システムでは、ユーザーを照会すると DOMAIN¥USERNAME が返されます。この値がデータベースの定義と一致しない場合は、NT 認証で ORA-1017 のエラーが発生します。

4. SQL*Plus を使用してデータベースに接続することにより、Windows NT 認証が正しく設定されているかどうかをテストします。ユーザー名には「/」と入力します。パスワード フィールドは空のままにし、ホスト文字列を正しく入力します。**OK** をクリックします。ログオンできた場合は、Windows NT 認証が正しく機能していることを意味します。

Oracle データベース認証

 **重要:** Oracle で Silk Test Workbench のユーザーを設定する場合、リモートで認証されるユーザーに加え、実際のテーブルを格納するためのスキーマも作成する必要があります。スキーマはパスワード認証されます。詳細については、「[パスワード認証スキーマの作成](#)」を参照してください。

ユーザーの作成には SQL*Plus を使用します。SQL*Plus を使い慣れていない場合は、Oracle のマニュアルを参照してください。以下の手順を実行するユーザーは、ORA_DBA グループのメンバーであり、SYSDBA 権限を持っている必要があります。

Oracle データベースにアクセスできるユーザーを指定するには、以下を実行します。

1. SQL*Plus または SQL*Plus Worksheet を起動します。
2. SYSTEM としてログオンし、SYSDBA として接続します。
3. 各ユーザーを作成するコマンドを "USERNAME" IDENTIFIED BY "ORACLEPASSWORD" として入力し、各ユーザーの接続権限を "USERNAME" として付与します。USERNAME は大文字で入力し、二重引用符で囲みます。DOMAIN¥USERNAME は各ユーザーのユーザー名です。例：


```
SQL> create user "JEFF" IDENTIFIED BY "ORACLEPASSWORD";  
SQL> grant connect, resource to "JEFF";
```

 **注:** SELECT_ANY_SEQUENCE、SELECT_CATALOG_ROLE、SELECT_ANY_DICTIONARY もユーザーに付与する必要があります。例：

```
SQL> grant select any sequence to "JEFF";  
SQL> grant select_catalog_role to "JEFF";  
SQL> grant select any dictionary to "JEFF";
```

パスワード認証スキーマの作成

Silk Test Workbench テーブルが格納されるパスワード認証スキーマを設定するには、以下のステップを実行します。

 **注:** SQL シリーズにおける Oracle の文字数制限のため、パスワード認証スキーマは 5 文字以下で指定する必要があります。

1. SQL*Plus または SQL*Plus Worksheet を起動します。
2. SYSTEM としてログオンし、SYSDBA として接続します。
3. 各ユーザーを作成するコマンドを "USERNAME" IDENTIFIED BY "ORACLEPASSWORD" として入力し、各ユーザーの接続権限を "USERNAME" として付与します。

USERNAME は大文字で入力し、二重引用符で囲みます。USERNAME は各ユーザーのユーザー名です。例：

```
SQL> create user "TOM" IDENTIFIED BY "ORACLEPASSWORD";  
SQL> grant connect, resource to "TOM";
```

Oracle データベースの準備

Silk Test Workbench で使用する新しい Oracle データベースを準備する必要があります。このセクションでは、**データベース メンテナンス** ユーティリティを使用してデータベースに Silk Test Workbench テーブルを入力する方法について説明します。

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。**データベース メンテナンス** ユーティリティが起動されます。
2. **ファイル > 新規作成 > Oracle** をクリックします。**Oracle データソースの接続** ダイアログ ボックスが開きます。
3. 新しい Oracle データソースの名前を入力するか、**参照** をクリックして **ODBC データソースの選択** ダイアログ ボックスからデータソース名を選択します。
4. 該当するボックスに、Oracle ユーザー ID の名前、パスワードを入力し、**作成** をクリックします。**スキーマ** ボックスには、ユーザー ID に入力した値が自動的に入力されます。

これで、Silk Test Workbench を起動してログオンする準備が整いました。


ドメインなしでの SQL Server または Oracle データベースの設定

Silk Test Workbench で使用する Oracle または SQL Server のデータベースを設定する際は、ドメイン内の認証を使用することを強くお勧めします。ただし、ドメインに所属しない場合でも、これらのデータベースを設定することは可能です。

このセクションでは、ドメインを使用して SQL Server または Oracle データベースを設定する具体的な方法について説明します。

ドメインなしでのユーザーの作成

ドメインがない場合に SQL Server または Oracle データベースを設定するには、最初にクライアント コンピュータとサーバー コンピュータの両方でユーザー アカウントを作成する必要があります。

 **注:** この手順は Windows NT 認証を使用する場合にのみ適用されます。

1. Windows の **コントロール パネル** にアクセスし、**管理ツール > コンピュータの管理** を選択します。**コンピュータの管理** ユーティリティが開きます。
2. 左側のペインで、**ローカル ユーザーとグループ** をクリックします。
3. **ローカル ユーザーとグループ** で、**ユーザー** フォルダを右クリックし、**新しいユーザー** を選択します。**新しいユーザー** ダイアログ ボックスが開きます。
4. **ユーザー名** テキスト ボックスに有効なユーザー名を入力し、**フル ネーム** テキスト ボックスにそのユーザーのフル ネームを入力します。
5. **パスワード** と **パスワードの確認入力** の両方のテキスト ボックスに有効なパスワードを入力し、**作成** をクリックします。
6. 各クライアント コンピュータでそのユーザー名とパスワードを使用して、上記の手順を繰り返します。

ドメインなしでの SQL Server の設定

ドメインがない場合に SQL Server を設定するには：

- Windows の **コントロール パネル** にアクセスし、**管理ツール > コンピュータの管理** を選択します。
コンピュータの管理 ユーティリティが開きます。
- 左側のペインで、**ローカル ユーザーとグループ** を展開し、**グループ** をクリックします。
- 右側のペインで、**Users** をダブルクリックします。
Users のプロパティ ダイアログ ボックスが開きます。
- 追加** をクリックします。
ユーザー、コンピュータ、またはグループの選択 ダイアログ ボックスが開きます。
- 詳細設定** をクリックし、**共通クエリ** タブを使用してユーザーを検索します。
- 今すぐ検索** をクリックして、リストからユーザーを選択します。
または、**選択するオブジェクト名を入力してください** テキスト ボックスをクリックし、アクセス権を付与する対象ユーザーの [user ID] を入力します。[user ID] :
- OK** をクリックしてユーザー情報を保存し、**Users のプロパティ** ダイアログ ボックスに戻ります。
- 上記のステップを繰り返して、その他のユーザー アカウントを作成します。
- OK** をクリックします。

ドメインなしでの Oracle データベースの設定

ドメインがない場合に Oracle を設定するには、個別のユーザー アカウントに対してサーバー アクセスを作成する手順に従います。ただし、ドメインなしでユーザーを作成する手順で作成したユーザー名を置き換えます。

- 選択するオブジェクト名を入力してください** テキスト ボックスに、[domain name]¥[user ID] や [domain name]¥[group name] ではなくユーザー名を入力します。

コンピュータの管理 ユーティリティが表示されます。

- ユーザーを設定します。

SQL*Plus を使用してユーザーを作成する場合は、以下の例に示すように [domain name¥user name] を [computer name¥user name] に置き換えます。

```
SQL> create user "FH0002RED¥USER1" IDENTIFIED EXTERNALLY;
SQL> grant connect, resource to "FH0002RED¥USER1";
```



注:

[computer name¥user name] の場合、大文字を使用して引用符で囲みます。

データ ソース名の作成

Silk Test Workbench 用にセットアップしたデータベースにアクセスするために、データベースに接続する各マシン上で、ODBC ユーティリティを使用して Silk Test Workbench データベースに対するデータ ソース名 (DSN) を作成します。



注: このステップは、Silk Test Workbench インストーラによってインストールされたデフォルトの Access データベースには適用されません。これらのデータベースでは、データ ソース名が自動的に作成されます。


Access データベース用のデータ ソース名の作成

Access データベースに接続する各クライアント コンピュータで、以下の手順に従って DSN を設定します。

- Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データ ソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データ ソース (32 ビット)** をクリックします。**ODBC データ ソース アドミニストレータ** ダイアログ ボックスが開きます。

2. **システム DSN** タブをクリックし、**追加** をクリックします。 **データソースの新規作成** ダイアログボックスが開きます。このダイアログボックスには、使用可能な ODBC ドライバがリストされます。
3. ドライバのリストから **Microsoft Access Driver** を選択し、**完了** をクリックします。 **ODBC Microsoft Access セットアップ** ダイアログボックスが開きます。
4. **データソース名** テキストボックスに、データソース名を入力します。
この名前はログオンダイアログボックスでデータベース名として表示されるため、論理名を入力します。
5. **説明** テキストボックスに、データソースの説明を入力します。
たとえば、「Silk Test Workbench データベースへの接続」です。
6. **選択** をクリックして、Access データベースを参照します。
7. **ODBC Microsoft Access セットアップ** ダイアログボックスで、**OK** をクリックします。
8. **ODBC アドミニストレータ** ダイアログボックスで、**OK** をクリックしてプロセスを完了します。

SQL Server データベース用のデータソース名の作成

 **注:** Silk Test Workbench では、64 ビットの DSN は使用できません。64 ビットマシンで DSN を作成するには、Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データソース (32 ビット)** をクリックして、32 ビット DSN を作成します。C:\%WINDOWS%\SysWOW64\odbcad32.exe にある WOW64 ツールを使用することもできます。

SQL Server データベースに接続する各クライアントコンピュータで、以下の手順に従って DSN を設定します。

1. Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データソース (32 ビット)** をクリックします。 **ODBC データソース アドミニストレータ** ダイアログボックスが開きます。
2. **システム DSN** タブをクリックし、**追加** をクリックします。 **データソースの新規作成** ダイアログボックスが開きます。このダイアログボックスには、使用可能な ODBC ドライバがリストされます。
3. ドライバのリストから **DBC Driver 11 for SQL Server** を選択し、**完了** をクリックします。 **SQL Server に接続するための新規データソースを作成する** ダイアログボックスが開きます。
4. **名前** ボックスに、データソース名を入力します。
この名前はログオン画面でデータベース名として表示されるため、論理名を入力します。
5. **説明** テキストボックスに、データソースの説明を入力します。
たとえば、「Silk Test Workbench データベースへの接続」です。
6. サーバー名を **サーバー** テキストボックスに入力するか、ドロップダウンリストから選択します。
7. **次へ** をクリックします。
8. 以下のいずれかを実行します。
 - DSN を使用して SQL Server データベースにアクセスするユーザーが非ネイティブのアカウント情報によって接続している場合、**ネットワークへのログイン ID で、Windows の認証メカニズムを使う** オプションを選択します。
 - DSN を使用して SQL Server データベースにアクセスするユーザーがネイティブ認証によって接続している場合、**ユーザーが入力する SQL Server 用のログイン ID とパスワードを使う** オプションを選択します。 **SQL Server に接続して追加の構成オプションの既定設定を取得する** チェックボックスがオンになっていることを確認してください。ユーザーのネイティブ SQL Server ログオン情報を、**ログイン ID とパスワード** の各テキストボックスに入力します。
9. **次へ** をクリックします。
- 10 **既定のデータベースを以下のものに変更する** チェックボックスをオンにして、リストからデータベースの名前を選択します。

11. その他のエントリはすべてそのままにし、ウィザードの最後のパネルが表示されるまで **次へ** をクリックします。

12. デフォルトをそのまま使用し、**完了** をクリックします。


13. 接続をテストするには、**データソースのテスト** をクリックします。**SQL Server ODBC データソーステスト** ダイアログボックスが開きます。

14. **OK** をクリックします。

ODBC データソースアドミニストレータ ダイアログボックスが再表示されます。新しく作成したデータソースが **システムデータソース** リストに表示されます。


15. **OK** をクリックします。

Oracle データベース用のデータソース名の作成


 **注:** Silk Test Workbench では、64 ビットの DSN は使用できません。64 ビットマシンで DSN を作成するには、Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データソース (32 ビット)** をクリックして、32 ビット DSN を作成します。C:\WINDOWS\SysWOW64\odbcad32.exe にある WOW64 ツールを使用することもできます。

Oracle データベースに接続する各クライアントコンピュータで、以下の手順に従って DSN を設定します。

1. Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データソース (32 ビット)** をクリックします。**ODBC データソースアドミニストレータ** ダイアログボックスが開きます。
2. **システム DSN** タブをクリックし、**追加** をクリックします。**データソースの新規作成** ダイアログボックスが開きます。このダイアログボックスには、使用可能な ODBC ドライバがリストされます。
3. ドライバのリストから、使用する Oracle データベースのバージョンに適した Oracle ODBC ドライバ (Microsoft ODBC for Oracle ドライバではない) を選択し、**完了** をクリックします。**Oracle ODBC ドライバ構成** ダイアログボックスが開きます。
4. **Oracle データソース名** テキストボックスに、データソース名を入力します。
この名前はログオンダイアログボックスでデータベース名として表示されるため、論理名を入力します。
5. **説明** テキストボックスに、データソースの説明を入力します。
たとえば、「Silk Test Workbench データベースへの接続」です。
6. **TNS サービス名** テキストボックスで、接続するデータベースのサービス名を ORACLE_TNS.DOMAIN の形式で選択してください。大抵の場合、ORACLE_TNS は、作成時にデータベースに割り当てられた Oracle SID であり、DOMAIN は、接続するドメインになります。

 **注:** ODBC を使用して Silk Test Workbench に接続するには、TNS サービス名は必須です。

7. Oracle 認証を使用する場合は、接続先のデータベースの有効なユーザー ID を **ユーザー ID** テキストボックスに入力します。それ以外の場合は、**ユーザー ID** テキストボックスは空のままにします。
8. 接続をテストするには、**接続テスト** をクリックします。

 **注:** Oracle データベース認証を使用する場合、ユーザー名を入力してから、パスワードを入力します。Oracle OS 認証では、ユーザー名を指定しないでください。

正常に接続されたことを示すメッセージが表示されます。

9. **OK** をクリックします。


データベースへ接続する

Silk Test Workbench をインストールしてデータベースを構成したあと、データベース接続を作成する必要があります。この接続によって、Silk Test Workbench データベースと Silk Test Workbench インストー


ルとの関係を定義します。このセクションでは、ログオンしてデータベース接続を作成する方法について説明するトピックを示します。

ログオン

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > クライアント > Silk Test Workbench**、または (Microsoft Windows 10) **スタート > Silk > Silk Test Workbench** をクリックします。Silk Test Workbench が起動され、**SilkTest Workbench にログイン** ダイアログ ボックスが表示されます。
2. **認証** リストボックスから、**Silk Test Workbench** 認証、または **Windows** 認証のいずれかを使用する方を選択します。

 **注:** **Windows** 認証を使用できるユーザーは、**ツール > 管理** を使用して Silk Test Workbench ユーザーに追加されたユーザーだけです。

3. **Silk Test Workbench** 認証を使用している場合、**ユーザー名** フィールドにユーザー名を入力し、**パスワード** フィールドにパスワードを入力します。**Windows** 認証を使用している場合は、ユーザー名とパスワードを入力する必要はありません。

 **注:** Silk Test Workbench をはじめて使用する場合は、管理者権限を持つユーザー名とパスワードが必要になる場合があります。デフォルトのユーザー名は Admin で、デフォルトのパスワードは admin です。不正アクセスを防ぐため、このパスワードはログオンのあとで変更してください。ログオンパスワードはいつでも変更できます。

4. **データベース** リストから、使用するデータベースを選択します。
データベースが **データベース** リストに表示されるようにするには、まずデータベース接続を使用できるように構成する必要があります。データベース接続を構成するには、**ログオン** ダイアログ ボックスで **構成** ボタンをクリックします。
5. **OK** をクリックします。

Silk Test Workbench データベースに初めてログインすると、*Project1* という名前のデフォルトプロジェクトがデータベースに作成されます。デフォルトでは、作成する資産は、このプロジェクトに保存されます。プロジェクト名は、**プロジェクトの編集** をクリックすると変更できます。

開始画面 が開きます。Silk Test Workbench に始めてログインすると、Silk Test Workbench は *Project1* または最初に利用可能な非グローバル オブジェクトをアクティブ オブジェクトとして選択します。

ログオンパスワードの変更

ログオンパスワードを変更するには、以下の手順に従います。

1. **ツール > パスワードの変更** をクリックします。**パスワードの変更** ダイアログ ボックスが開きます。現在のユーザーのログオン名が、ダイアログ ボックスのタイトルバーに表示されます。
2. パスワードを変更し、**OK** をクリックします。このユーザー ID を使って次回ログオンしたとき、変更が有効になります。


データベース接続の構成


Silk Test Workbench でデータベースを使用する前に、データベース接続を構成する必要があります。構成処理には、**データベース接続の構成** ダイアログ ボックスを使用した以下のタスクの実行が含まれています。

- データベース接続を開始するために必要なデータベース接続データを指定します。
- データベースへの接続を検証します。
- データベース接続のデータを保存します。

既存のすべての構成済みデータベース接続は、Silk Test Workbench ダイアログ ボックスにある **データベース** リストに表示されます。

データベース接続の構成 ダイアログ ボックスは、既存の構成済みデータベース接続の表示、編集、および削除に使用することもできます。

 **注:** Silk Test Workbench で使用するデータベース接続を構成する前に、データベース インスタンスと ODBC データ ソース名 (DSN) を作成および構成しておく必要があります。

 **注:** Silk Test Workbench では、64 ビットの DSN は使用できません。64 ビット マシンで DSN を作成するには、Microsoft Windows 7 の場合は **スタート > Silk > Silk Test > 管理 > データ ソース (ODBC)** を、Microsoft Windows 10 の場合は **スタート > Windows 管理ツール > ODBC データ ソース (32 ビット)** をクリックして、32 ビット DSN を作成します。C:\¥WINDOWS ¥SysWOW64¥odbcad32.exe にある WOW64 ツールを使用することもできます。

1. Silk Test Workbench に**ログイン** ダイアログ ボックスで、**構成** をクリックします。 **データベース接続の構成** ダイアログ ボックスが開きます。
2. **データ ソースの種類** セクションで、データ ソースの種類を選択します。
3. **データ ソースの選択** リストで、使用可能な DSN のリストから選択します。
4. 選択した DSN を Silk Test Workbench に**ログイン** ダイアログ ボックスの **データベース** リストに表示される構成済みデータベース接続のリストに追加するには、**Silk Test Workbench データベースとして使用** チェック ボックスをオンにします。
データベース接続をはじめて構成するときは、必ずこのチェック ボックスをオンにしてください。既存の構成済みデータベース接続に対して **Silk Test Workbench データベースとして使用** チェック ボックスをオフにして **適用** をクリックすると、選択した DSN がダイアログ ボックスの **データベース** リストから削除されます。Silk Test Workbench は関連するデータベース接続のデータを保持しません。
5. **データベースの設定** セクションで、選択した DSN に対して適切な情報を指定します。Access では、**データベース** ボックスは読み取り専用で、データベース ファイルの場所を示します。この値は ODBC DSN から直接読み取られます。他の **データベースの設定** テキスト ボックスは、Access には適用されず、無効になっています。Oracle では、**サーバー** テキスト ボックスは読み取り専用で、データベース名のみが表示されます。この値は ODBC DSN から直接読み取られます。
6. **データベース** ボックスに、適切な値を入力します。SQL Server/MSDE では、**データベース** テキスト ボックスは編集可能で、適切なデータベース名を入力することができます。
7. **所有者** ボックスに、適切な値を入力します。
8. 認証の種類を選択します。
認証 オプションは、Oracle データベースまたは SQL Server/MSDE データベースにのみ適用されます。Windows NT 認証を使用するか、データベースのネイティブ認証機能を使って認証するかを選択できます。データベース認証を選択する場合、適切なテキスト ボックスにユーザー名とパスワードを入力する必要があります。
9. 省略可能：現在選択されているデータベース構成を暗号化ファイルに保存する場合は、**エクスポート** をクリックします。
このボタンは、現在選択されているデータベースが Silk Test Workbench データベースとして構成されている場合にのみ、有効になります。
10. 省略可能：データベースの構成情報を暗号化ファイルから追加する場合は、**インポート** をクリックします。
11. **検証** をクリックして、指定したデータベース接続データを使用してデータベースに接続できることを確認します。
12. **適用** をクリックして、データベース接続データを保存します。
13. **閉じる** をクリックして、ダイアログ ボックスを閉じます。

データベース構成の共有

データベース構成を共有して、複数のマシン上で、同じデータベースを同じ設定で構成できます。

1. あるマシン上でデータベースを構成します。
詳細については、「[Silk Test Workbench データベースの構成](#)」を参照してください。

2. データベース接続を構成します。
詳細については、「[データベース接続の構成](#)」を参照してください。
3. **データベース接続の構成** ダイアログを使って、データベース構成ファイルをエクスポートします。
詳細については、「[データベース接続の構成](#)」を参照してください。
4. データベース構成を共有するマシンそれぞれで、以下の手順を実行します。
 - a) ファイル エクスプローラを開きます。
 - b) %PROGRAMDATA%\Silk\SilkTest Workbench ディレクトリに移動します。
 - c) Automatic Database Configurations という名前の新しいサブフォルダーを作成します。
 - d) データベース構成ファイルをこの新しいフォルダーにコピーします。コピーしたマシン上で、次に Silk Test Workbench を開始したときに、DSN が自動的に作成され、データベースへの構成情報が保存されます。新しいデータベースは、Silk Test Workbench のログイン ダイアログの選択肢に表示されます。

共有したデータベース構成には、次のルールが適用されます。

- 共有したデータベース構成は、**データベース接続の構成** ダイアログで編集できません。
- データベース構成情報を変更すると、既存のデータベース構成ファイルを、新しいバージョンのファイルで置き換えることができます。Silk Test Workbench がファイルの変更を検出すると、データベース構成は適切に更新されます。
- 特定のマシン上でデータベース構成がなくなっただけの場合は、そのファイルを %PROGRAMDATA%\Silk\SilkTest Workbench\Automatic Database Configurations ディレクトリから削除します。このマシン上で、次に Silk Test Workbench を開始したときに、DSN とデータベースへの構成情報が削除されます。

データベース メンテナンス

データベース メンテナンス ユーティリティは、データベース スキーマの更新、データベースのレコードのロック解除、データベースのコンパクト化など、いくつかのメンテナンス タスクを実行します。

メンテナンス タスクを実行する前に、Silk Test Workbench データベース接続を構成する必要があります。また、Silk Test インストーラを使用せずにインストールされたデータベースでは、**データベース メンテナンス** ユーティリティを使用して新しい Silk Test Workbench ODBC データベースを作成する前に、システム管理者が新しい ODBC データソースを作成する必要があります。詳細については、『*Silk Test インストール ガイド*』を参照してください。

Silk Test Workbench データベースを開く

すでに作成されている Access、SQL Server、または Oracle Silk Test Workbench データベースを開いて、データベース スキーマの更新、データベースのレコードのロック解除、データベースのコンパクト化などのメンテナンス タスクを実行できます。

Microsoft SQL Server データベースを開く

データベース メンテナンス ユーティリティを起動して Microsoft SQL Server データベースを開くには、以下の手順に従います。

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。 **データベース メンテナンス** ユーティリティが起動されます。
2. **ファイル > データベースを開く > SQL Server** をクリックします。 **SQL Server データソースの接続** ダイアログ ボックスが開きます。
3. **参照** をクリックします。 **ODBC データソースの選択** ダイアログ ボックスが開きます。

4. 目的の Microsoft SQL Server DSN を選択して、**選択** をクリックします。 **SQL Server データソースの接続** ダイアログ ボックスが開き、**SQL Server データソース名** テキスト ボックスに選択した DSN が表示されます。



注: デフォルトの Silk Test Microsoft SQL Server 2008 Express データベースの DSN は Silk Test_SQL です。

5. **所有者**、**ユーザー ID**、および **パスワード** テキスト ボックスに、Silk Test Microsoft SQL Server データベース インスタンスに関する適切な情報を入力します。



注: デフォルトの Silk Test Microsoft SQL Server 2008 Express データベースの所有者は Silk Test DB であり、管理者権限を持つユーザーのユーザー名は SilkTestAdmin、パスワードは SilkTestAdmin です。

6. **開く** をクリックします。データベースが開き、**データベース メンテナンス ユーティリティの ツール** メニュー オプションが使用可能になります。選択したデータベースに現在接続しているユーザーがいる場合は、メッセージが表示されます。詳細については、「接続ユーザーがいるデータベースのメンテナンスの実行」を参照してください。

Oracle データベースを開く

データベース メンテナンス ユーティリティ を起動して Oracle データベースを開くには、以下の手順に従います。

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。 **データベース メンテナンス ユーティリティ** が起動されます。
2. **ファイル > データベースを開く > Oracle** をクリックします。
Oracle データソースの接続 ダイアログ ボックスが開きます。
3. **参照** をクリックします。 **ODBC データソースの選択** ダイアログ ボックスが開きます。
4. 目的の Oracle DSN を選択して、**選択** をクリックします。 **Oracle データソースの接続** ダイアログ ボックスが開き、**Oracle データソース名** テキスト ボックスに選択した DSN が表示されます。
5. **スキーマ**、**ユーザー ID**、および **パスワード** テキスト ボックスに、Silk Test Oracle データベース インスタンスに関する適切な情報を入力します。
6. **開く** をクリックします。データベースが開き、**データベース メンテナンス ユーティリティの ツール** メニュー オプションが使用可能になります。選択したデータベースに現在接続しているユーザーがいる場合は、Silk Test Workbench によってメッセージが表示されます。

詳細については、「接続ユーザーがいるデータベースのメンテナンスの実行」を参照してください。

Access データベースを開く

データベース メンテナンス ユーティリティ を起動して Microsoft Access データベースを開くには、以下の手順に従います。

1. (Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。 **データベース メンテナンス ユーティリティ** が起動されます。
2. **ファイル > データベースを開く > Access** をクリックします。 **Access データベースの接続** ダイアログ ボックスが開きます。
3. **参照** をクリックします。 **ODBC データソースの選択** ダイアログ ボックスが開きます。
4. 目的の Access DSN を選択して、**選択** をクリックします。
5. **開く** をクリックします。

詳細については、「接続ユーザーがいるデータベースのメンテナンスの実行」を参照してください。

データベースが開き、**データベース メンテナンス ユーティリティの ツール** メニュー オプションが使用可能になります。選択したデータベースに現在接続しているユーザーがいる場合は、Silk Test Workbench によってメッセージが表示されます。

データベース バージョンの更新

Silk Test Workbench を最新のバージョンにアップグレードする場合、既存のデータベースを変換して新しいソフトウェアと互換性を持たせる必要があります。Silk Test Workbench では、データベースを読み込んだときに自動的にデータベーススキーマのバージョンがチェックされ、データベースのアップグレードが必要な場合は警告が表示されます。この警告は、データベーススキーマが現在の Silk Test Workbench スキーマと一致しない場合には必ず表示されます。データベースは、**データベース メンテナンス ユーティリティ**を使用して変換できます。

変換速度は、変換するデータ量に関係しています。大量のデータを含むデータベースは変換に時間がかかる場合があります。変換後、現在のデータベースバージョン番号が表示されます。



注: データベースを新しいスキーマに変換する場合、Silk Test Workbench の古いバージョンでデータベースにアクセスできなくなります。つまり、すべての Silk Test Workbench ユーザーが新しいバージョンの Silk Test Workbench にアップグレードする必要があります。

1. 必要に応じて、Silk Test Workbench のログイン要求を取り消し、ユーザーがデータベースにログインしないようにします。
2. **データベース メンテナンス ユーティリティ**を起動して、データベースを開きます。**データベース メンテナンス ユーティリティ**のメインウィンドウが表示されます。
3. **ファイル > データベースを開く** をクリックし、データベースの種類を選択します。**データベース接続** ダイアログ ボックスが表示されます。
4. データベース名とパスワードを入力するか、**参照** をクリックします。
5. **開く** をクリックします。
データベースの更新が必要な場合は、**ツール > 変換** をクリックしてデータベースを更新するよう求めるメッセージ ボックスが表示されます。
6. **OK** をクリックします。データベースが開き、**データベース メンテナンス ユーティリティ**の**ツール**メニュー オプションが有効になります。
7. **変換** をクリックします。**データベースの変換** ダイアログ ボックスが開き、変換前に現在のデータベースをバックアップするかどうかを尋ねられます。データベースのバックアップを選択します。
8. **今すぐ変換** をクリックして、データベースを現在のスキーマバージョンに更新します。
9. 変換が完了したら、**閉じる** をクリックします。

データベースレコードのコピー

データベース メンテナンス ユーティリティでは、1 つのデータベースから別のデータベースにレコードをコピーすることによって、データベースの複製コピーを作成できます。たとえば、あるサイトで Silk Test Workbench 資産リポジトリとして Microsoft Access を使用しており、Silk Test Workbench のデータを Oracle データベースに格納する場合は、**データベース メンテナンス ユーティリティ**を使用して、Access データベースに格納されているデータを Oracle データベースにコピーします。コピー ユーティリティを使用すると、Silk Test Workbench のすべてのデータがターゲット データベースで読み取り可能な形式に自動的にコピーされます。

ネットワーク データベースではコピー操作を実行しないでください。ネットワーク データベースをローカルのワークステーションにコピーし、コピー機能を実行してからネットワークの元の場所にデータベースを戻してください。これによって、ネットワークの障害や中断によるコピー作業への影響を防ぐことができます。

1. ユーザーがデータベースに接続していないことを確認してください。
2. **データベース メンテナンス ユーティリティ**を起動して、コピー元データベースを開きます。
3. **コピー ツールバー** ボタンをクリックします。**コピー先** ダイアログ ボックスが開きます。
4. コピー先データベースの種類を選択して、**OK** をクリックします。

- Access データベースへのコピー
 1. **DSN** テキスト ボックスで、**参照** をクリックして、コピー先データベースの DSN を選択します。**Access** テキスト ボックスに、データベースのパスと名前が表示されます。
 2. **コピー** をクリックします。**テーブルのコピー中** ダイアログ ボックスが開き、データベース コピー処理の進行状況が表示されます。
- Microsoft SQL Server データベースのコピー
 1. Microsoft SQL Server データ ソースの名前を入力するか、または **参照** をクリックしてデータ ソース名を選択します。所有者名はデフォルトで dbo になります。
 2. ユーザー ID を入力します。ユーザーは、dbo 権限を持っている必要があります。
 3. データベース サーバー上の Silk Test Workbench データベース インスタンスのパスワードを入力します。このユーザー ID とパスワードは、データベース サーバー上の Silk Test Workbench データベース インスタンスへの接続に使用されます。
 4. **コピー** をクリックします。所有者名、ユーザー ID、パスワードがデータベース サーバーで承認された場合は、**データソースの作成** ダイアログ ボックスが開き、データベース コピー処理の進行状況が表示されます。
- Oracle データベースへのコピー
 1. 新しい Oracle データ ソースの名前を入力するか、または **参照** をクリックして既存のデータ ソース名を選択します。**スキーマ** および **ユーザー ID** テキスト ボックスのデフォルトは **システム** であるため、デフォルト設定を使用する場合は、システム管理者または該当するパスワードを把握しているユーザーのみがメンテナンスを実行できます。
 2. **コピー** をクリックします。スキーマ名、ユーザー ID、パスワードがデータベース サーバーで承認された場合は、**データソースの作成** ダイアログ ボックスが開き、コピー処理の進行状況が表示されます。**データベース メンテナンス** ユーティリティによって、Silk Test Workbench データベース インスタンスへの必要な Silk Test Workbench テーブルのコピーが開始されます。

データベースのレコードのロック解除

資産が開いているとき、他のユーザーが同じ資産を同時に編集しないように、そのレコードはロックされます。リソースを保存するかシステムを終了すると、ロックが解除されます。システムがクラッシュした場合は、使用していたレコードが起動時にロックされたままになっている可能性があります。システムのクラッシュ後に、誤ってロックされたままになっているレコードがあると、Silk Test Workbench はロック解除を試みます。

プログラムを使用する前に、すべてのユーザーがシステムからログオフしていることを確認してください。接続しているユーザーがいる場合は、メッセージが表示されます。

1. **データベース メンテナンス** ユーティリティを起動して、データベースを開きます。
2. **レコードのロック解除** をクリックします。データベースに接続しているユーザーがいる場合は、データベース メンテナンス ユーティリティでデータベースのロック ダウンの際に接続ユーザーが発見されました。というメッセージが表示されます。
3. **OK** をクリックします。**接続ユーザー** ダイアログ ボックスが開きます。**ユーザー** リストに、現在接続しているユーザーのマシン名が表示されます。
4. **送信** をクリックして、これらのユーザーにログオフするよう勧告する通知を送ります。これらのユーザーがログオフしたら、ステップ 1 と 2 を繰り返し、データベース レコードのロックを解除します。



注: 接続しているユーザーにメッセージを送信するには、オペレーティング システムの Messenger サービスを有効にする必要があります。Messenger サービスを有効にするには、**コントロール パネル > 管理ツール > サービス** をクリックし、**Messenger** を選択して、サービスを開始します。

Access データベースのコンパクト化

データベースが日常的に使用されている場合は、最適なパフォーマンスを得るために定期的にデータベースのコンパクト化を行います。リソースの更新や削除を繰り返すうちに、データベース ファイルの断片化

が進みます。断片化はディスクスペースを浪費し、パフォーマンスを低下させることがあります。Access データベースの場合、この手順を実行するまでは、データベースから資産が削除されてもファイルサイズの減少は登録されません。**データベース メンテナンス ユーティリティ**を使用すると、不要なレコードを削除してデータをディスクの連続的な領域に書き込み直すことで、Microsoft Access データベースをコンパクト化できます。

ネットワーク データベースにはデータベースのコンパクト化処理を実行しないでください。ネットワーク データベースをローカルのワークステーションにコピーしてからデータベースのコンパクト化処理を行い、作業が終わったらネットワークの元の場所に戻してください。これによって、ネットワークの障害や中断によるデータベース メンテナンス作業への影響を防ぐことができます。

1. すべてのユーザーがシステムからログオフしていることを確認し、Silk Test Workbench を終了します。
2. **データベース メンテナンス ユーティリティ**を起動して、データベースを開きます。
3. **コンパクト化**をクリックします。確認のためのダイアログ ボックスが開きます。
4. **はい**をクリックして続行します。処理が完了すると、データベースが正しくコンパクト化されたことを知らせるダイアログ ボックスが表示されます。
5. **OK**をクリックします。これで、データベースを使用できます。

データベース メンテナンス設定の構成

データベース メンテナンス ユーティリティでは、コンパクト化、変換、およびコピーの各処理中に使用される設定を構成できます。

メンテナンス システムの操作の設定 ダイアログ ボックスのバックアップ設定は、Microsoft Access データベースにのみ適用されます。Silk Test Workbench データベースとして使用する他の種類のデータベースに対してバックアップ設定を構成するには、各データベースの製品ドキュメントを参照してください。

データベースのコンパクト化や変換を行うときに、現在のバージョンをバックアップできます。**メンテナンス システムの操作の設定** ダイアログ ボックスを使用して、保持する最大バックアップ コピー数を指定します。コンパクト化または変換のバックアップ数を 3 つと指定した場合、データベースを変換またはコンパクト化するたびに、新しいバックアップが 1 つ作成されます。バックアップが 3 つ作成されたあとにもう一度変換またはコンパクト化を実行すると、一番古いバックアップが削除されて新しいバックアップが追加されます。


トランザクション中にバッファに入れるレコードの数を選択します。トランザクションとは、パフォーマンスを向上させるために大量のデータをデータベースにサイクルで書き込むプロセスです。トランザクションがデータベースにコミットされるまで、各データベース処理がバッファに入れられます。ただし、1 つでも処理が失敗すると、そのトランザクションのすべての処理が失敗します。データをデータベースに保存する前にバッファに入れる処理の数を指定できます。

1. すべてのユーザーがシステムからログオフしていることを確認し、Silk Test Workbench を終了します。
2. **データベース メンテナンス ユーティリティ**を起動して、データベースを開きます。
3. **オプション > 構成**を選択します。**メンテナンス システムの操作の設定** ダイアログ ボックスが開きます。
4. コンパクト化および変換時に維持するバックアップ数の指定、コミットするレコード数の設定、および何秒経過するとクエリを失敗とみなすかを示す秒数の指定を行うには、**設定** タブでオプションをダブルクリックします。

接続ユーザーがいるデータベースのメンテナンスの実行

選択したデータベースに接続しているユーザーがいる場合、**データベース メンテナンス ユーティリティ**にメッセージが表示されます。また、**接続ユーザー** ダイアログ ボックスに接続ユーザーのリストが表示され、ここからこれらのユーザーに通知を送ることができます。デフォルトのメッセージを使用するか、または固有のメッセージを作成し、**送信**をクリックして、データベースからログオフするように接続ユーザーに通知します。

接続ユーザーのいるデータベースでメンテナンスを実行すると、ローカル マシンのシステムが不安定になるので、これは行わないでください。接続ユーザーがいなくなるまで待機する場合は、**一覧情報を更新** ボタンを使用して、接続ユーザーの最新のビューを表示できます。

 **注:** ユーザーがメッセージを受信するには、すべてのユーザーが Windows のメッセージ機能を実効にしている必要があります。それには、ユーザーがクライアント マシンの[サービス]にアクセスし、[Messenger]サービスをオンにします。メッセージの送信に使用される Windows のメカニズムによって複数のメッセージが送られる場合があります。

データベース整合性チェックの実行

データベース整合性チェックを正しく実行するには、Silk Test インストール ディレクトリの書き込み権が必要です。データベース整合性チェックに、Silk Test Workbench はこのディレクトリにデータベース整合性チェックの結果を書き込みます。書き込み件がない場合は、データベースがメンテナンスのためにロックされてしまう可能性があります。


データベース メンテナンス ユーティリティを使用すると、データベースの整合性を検証して、検出されたエラーを修正できます。

1. データベースを開き、**ツール > 整合性チェックの実行** を選択します。**資産の検証中** メッセージ ボックスが表示され、整合性チェックの進行状況が示されます。エラーがない場合は、メッセージ ボックスが閉じます。エラーが見つかったら、メッセージ ボックスが表示され、結果を確認するように求められます。
2. **はい** をクリックして、整合性チェックの結果を確認します。結果はデフォルトのテキスト エディタで表示されます。整合性チェックの結果が記録されたファイルは、Silk Test の logs ディレクトリ (通常は %LOCALAPPDATA%\Silk Test\logs) に IntegrityCheck.log という名前で保存されます。

データベースの膨張の抑制

ビジュアル テストの記録と再生中、Silk Test Workbench は記録と再生の各段階で、テスト アプリケーションの画面とコントロールをキャプチャし、格納します。多数のイメージやコントロールがキャプチャされる可能性があるため、データベースのサイズが急速に増えることがあります。

データベース サイズの増加を抑えるには、記録と再生の設定で **画面キャプチャ** および **コントロールのキャプチャ** を無効にします。詳細については、「記録出力オプションの設定」および「再生結果のビジュアル テスト オプションの設定」を参照してください。

 **注:** コントロールをキャプチャすると、**画面プレビュー** からコントロールを更新したり、記録したビジュアル テストにテスト アプリケーションのコントロールを挿入したりできます。また、この機能を無効にすると、選択したステップと関連するコントロールを示す強調ボックスが **画面プレビュー** で表示されなくなります。

たとえば、XML 形式にエクスポートしたとき、一般的なテスト アプリケーションの再生結果のサイズは、キャプチャ設定によって以下のようになります。

- 画面とコントロールの両方をキャプチャした場合：7.8MB
- 画面だけをキャプチャした場合：5.9MB
- 画面とコントロールのキャプチャを無効にした場合：382KB

プロジェクトの管理

プロジェクトは資産のコレクションであり、各プロジェクトではアクセス権限を各ユーザーに割り当てます。Silk Test Workbench 環境の管理者が行うタスクの 1 つに、プロジェクト リストの管理があります。資産を論理的に関連付けたプロジェクトとしてまとめて保管します。たとえば、特定のアプリケーションまたはビルドに関連付けられた資産をすべて一緒に、1 つのプロジェクトに保管できます。

管理者は、プロジェクトを作成して、Silk Test Workbench データベースに追加できます。作成した新しいプロジェクトのアクセス権は、各ユーザーに設定された [デフォルトの権限] が割り当てられます。資産を別々のプロジェクトに保管すると、セキュリティのレベルを変えることができます。たとえば、プロジェクト 1 はフル アクセス権、プロジェクト 2 は読み取り専用、プロジェクト 3 はアクセス不可、のように設定できます。

現在のユーザーには、アクセス権を割り当てられているプロジェクトの資産だけが表示されます。資産は、**アセットブラウザ** を使用して表示および管理します。



注: Silk Test Workbench が提供する機能を最適に使用するには、同じテストで複数のアプリケーションをテストする場合を除き、テストするアプリケーションごとに個別のプロジェクトを作成します。

Common プロジェクト

デフォルトのプロジェクトは Common プロジェクトです。これは、複数のプロジェクトで使用される資産の中央リポジトリとして機能します。Common プロジェクトの資産は、他のすべてのプロジェクトで使用できます。Common プロジェクトはデフォルトでアクティブ プロジェクトになっています。

アクティブ プロジェクト

新しく作成される資産は、アクティブ プロジェクトとして設定されているプロジェクトに追加されます。アクティブに追加された資産は、そのプロジェクト内と、アクティブ プロジェクトを参照しているすべてのプロジェクト内でのみ使用可能になります。



注: プロジェクトの資産は、同じプロジェクトまたは参照プロジェクトの他の資産にのみアクセスできます。

プロジェクトの追加

プロジェクトを Silk Test Workbench データベースに追加するには、管理者としてログインする必要があります。プロジェクトを作成する場合、新しいプロジェクトのアクセス権には、デフォルトのアクセス権が設定されます。

1. **ツール > 管理** を選択します。
または、**開始画面** の **タスク** ペインで **管理** をクリックします。
管理 ダイアログ ボックスが開きます。
2. **プロジェクト** タブをクリックします。
3. **追加** をクリックします。 **プロジェクトの追加** ダイアログ ボックスが開きます。
4. **名前** テキスト ボックスに、新しいプロジェクトの名前を入力します。
5. **OK** をクリックします。

プロジェクトの参照

プロジェクトの参照を利用すると、参照プロジェクトのコードや資産を再利用して、資産を体系化できます。さらに、大規模なチームでは、様々なチームがアプリケーションの異なる領域をテストしている場合などに、プロジェクトを参照することで、複数の場所で資産を共有することができます。

他のプロジェクトからプロジェクトを参照するには：

1. Silk Test Workbench メニューで、**ツール > 管理** を選択します。 **管理** ダイアログ ボックスが開きます。
2. **プロジェクト** タブを選択します。
3. 他のプロジェクトを参照するプロジェクトを選択します。
現在のユーザーがアクセス権を持たないプロジェクトには、取り消し線が引かれます。
4. **変更** をクリックします。

5. **利用可能** リストから参照するプロジェクトを選択します。
6. 矢印を使って、プロジェクトを **参照中** リストに移動します。

参照プロジェクトのすべての資産が、現在のプロジェクトから利用できるようになります。

プロジェクト参照の表示

プロジェクト参照を表示して、現在のプロジェクトが参照しているプロジェクトや、現在のプロジェクトを参照しているプロジェクトを確認できます。

1. Silk Test Workbench メニューから、**表示** > **プロジェクト参照** を選択します。
2. プロジェクト参照が開きます。
3. **参照** タブをクリックすると、現在のプロジェクトが参照しているプロジェクトが表示されます。
4. **参照元** タブをクリックすると、現在のプロジェクトを参照しているプロジェクトが表示されます。

グローバルプロジェクトの定義

プロジェクトをデータベース内のすべての他のプロジェクトから参照できるようにするには：

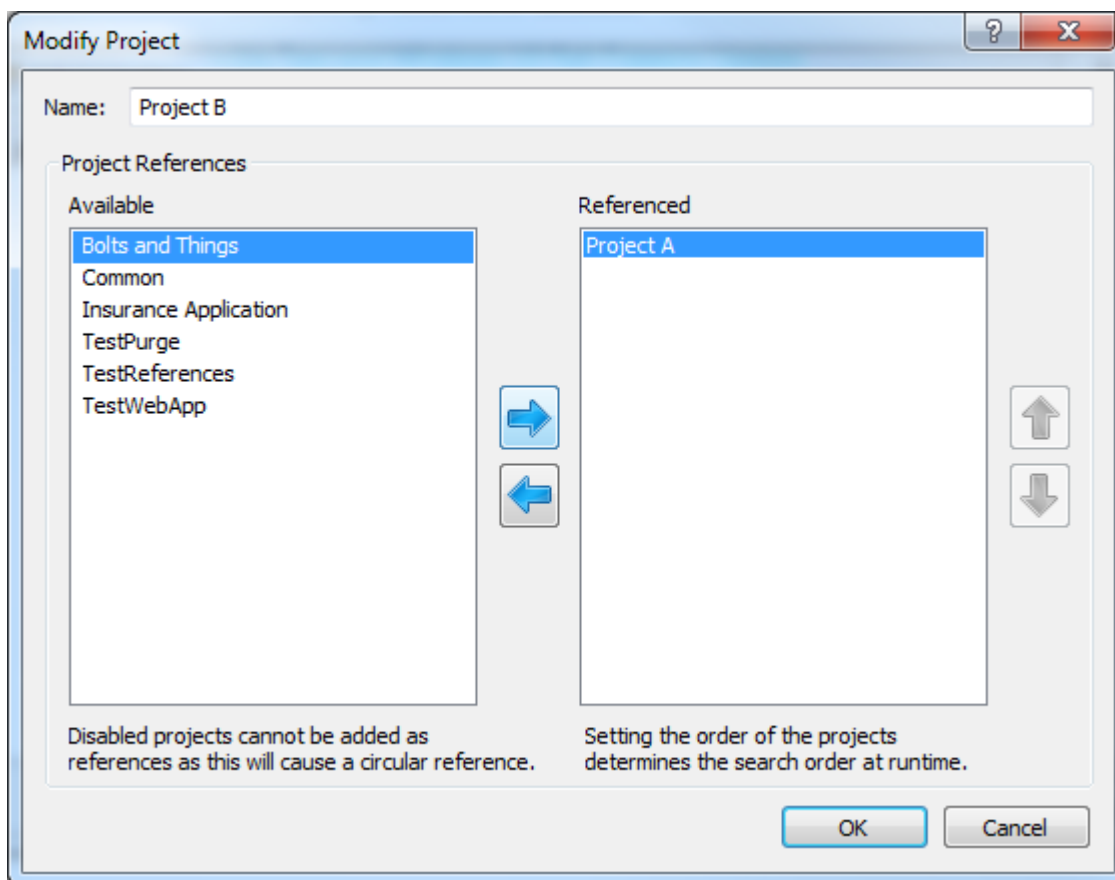
1. Silk Test Workbench メニューで、**ツール** > **管理** を選択します。 **管理** ダイアログ ボックスが開きます。
2. **プロジェクト** タブを選択します。
3. **グローバルプロジェクトの定義** をクリックします。 **グローバルプロジェクトの定義** ダイアログが表示されます。他のプロジェクトを参照していない、かつ他のプロジェクトから参照されていないプロジェクトが **利用可能** リストに表示されます。
4. **利用可能** リストからグローバルに参照するプロジェクトを選択します。
5. 矢印を使って、プロジェクトを **グローバルプロジェクト** リストに移動します。

グローバルプロジェクトのすべての資産が、データベース内のすべての他のプロジェクトから利用できるようになります。

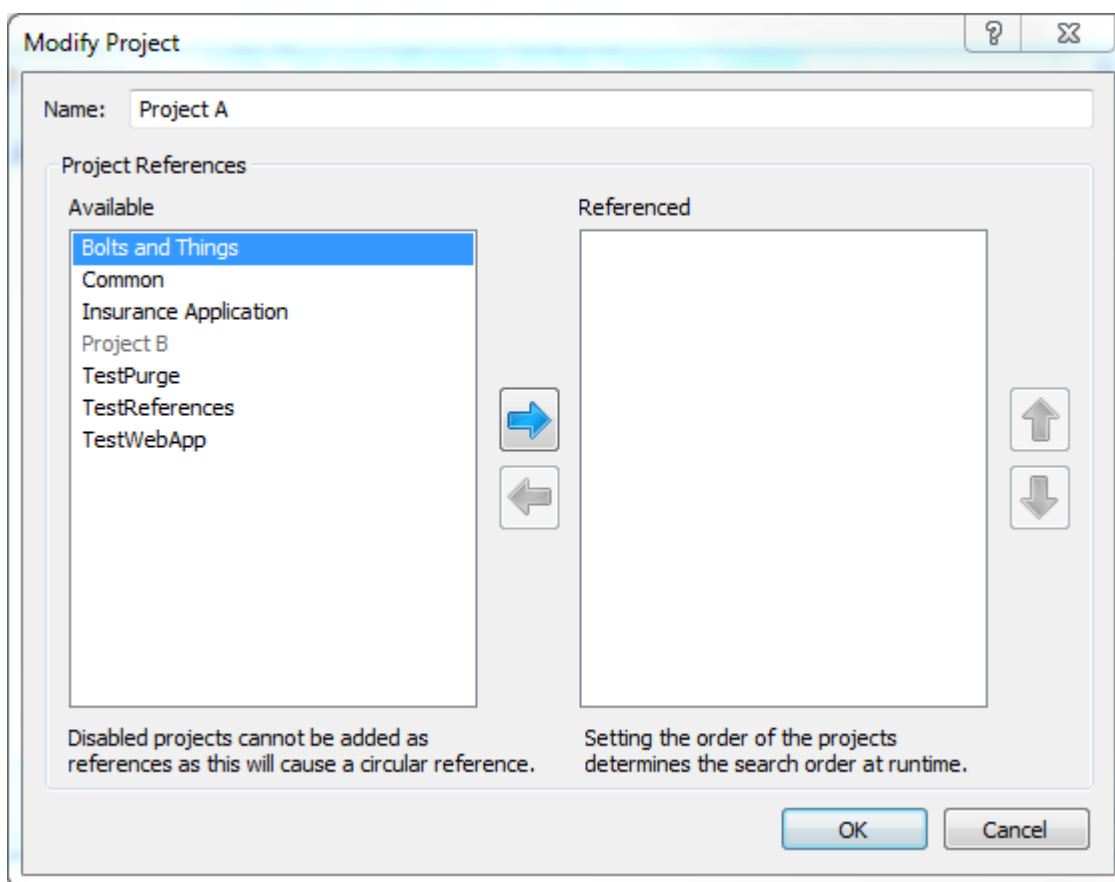
Silk Test Workbench がプロジェクトの循環参照を回避する方法

あるプロジェクトから他のプロジェクトへの新しい参照を追加するとき、Silk Test Workbench はプロジェクト間の参照が確実に循環しないことをチェックします。このトピックでは、Silk Test Workbench での循環参照のチェック方法について説明します。

循環参照の最も簡単な例として、2つのプロジェクト A とプロジェクト B があり、プロジェクト B がプロジェクト A を参照している状態を考えます。



プロジェクト A からプロジェクト B への参照を追加すると、循環参照が形成されます。このため、Silk Test Workbench では、この参照の追加は行えません。プロジェクト A を変更する場合、**プロジェクトの変更** ダイアログ ボックスの **利用可能** リストにプロジェクト B は表示されますが、無効になっています。




Silk Test Workbench では、これをチェーン参照のような複数の参照に対しても適用することで、プロジェクトの参照が循環することを回避します。

プロジェクト名の変更

Silk Test Workbench データベースのプロジェクト名を変更するには、管理者としてログインする必要があります。

1. ツール > 管理 を選択します。
または、開始画面のタスク ペインで 管理 をクリックします。
管理 ダイアログ ボックスが開きます。
2. プロジェクト タブをクリックします。
3. 名前を変更するプロジェクトを選択し、名前の変更 をクリックして、プロジェクト名の変更 ダイアログ ボックスを開きます。
4. 名前 テキスト ボックスに、プロジェクトの新しい名前を入力します。
5. OK をクリックします。
6. プロジェクトが他のプロジェクトから参照されている場合は、その処理方法を選択します。
 - プロジェクト名を変更して、参照を更新する場合は、このプロジェクトの名前を変更して参照を更新する をクリックします。名前を変更したプロジェクトを参照するプロジェクトへのスクリプト作成者アクセス許可があれば、そのプロジェクト内のすべての資産から、名前を変更したプロジェクトへの参照を探し出して、プロジェクトの新しい名前に参照を更新します。
 - 参照を更新せずにプロジェクト名を変更する場合は、このプロジェクトの名前のみ変更する をクリックします。

 **注:** 参照しているプロジェクトの資産は、正しく再生できなくなる可能性があります。

- プロジェクト名の変更をキャンセルする場合は、**キャンセル** をクリックします。
7. **OK** をクリックして **管理** ダイアログを閉じます。すべての参照しているプロジェクトの資産が更新されます。**管理** ダイアログを閉じずに複数のプロジェクトの名前を変更すると、参照しているプロジェクトの資産は、一度だけ更新されることとなります。

プロジェクトの複製

プロジェクトのベースラインを作成する場合など、プロジェクトを複製して、プロジェクトの完全なコピーを作成することができます。Silk Test Workbench データベースのプロジェクトを複製するには、管理者としてログインする必要があります。

1. **ツール** > **管理** を選択します。
または、**開始画面** の **タスク** ペインで **管理** をクリックします。
管理 ダイアログ ボックスが開きます。
2. **プロジェクト** タブをクリックします。
3. 複製するプロジェクトを選択し、**複製** をクリックします。**プロジェクトの複製** ダイアログ ボックスが表示されます。
4. 複製するプロジェクトの名前を **新しい名前** フィールドに指定します。
5. 省略可能：**すべての資産のバージョンをコピーする** チェック ボックスをオンにします。
デフォルトでは、このチェック ボックスはオフになっているため、元のプロジェクトの各資産のアクティブなバージョンだけがコピーされます。
6. プロジェクトが他のプロジェクトを参照している場合、各参照プロジェクトごとに **アクション** 列から適用するアクションを選択します。
 - 複製するプロジェクトから参照プロジェクトへの参照をそのまま保持する場合は、**参照の保持** を選択します。これは、デフォルトの設定です。
 - 参照プロジェクトも複製し、複製するプロジェクトからその複製した参照プロジェクトを参照する場合は、**アクション** 列で **プロジェクトの複製** を選択します。複製したプロジェクトの名前は、**新しい名前** 列で変更できます。
 - 子の参照プロジェクトを複製する場合は、親のプロジェクトも複製する必要があります。たとえば、プロジェクト B を参照するプロジェクト A を複製する際に、プロジェクト B がプロジェクト C を参照している場合は、プロジェクト C を複製するにはプロジェクト B も複製する必要があります。
7. **OK** をクリックします。



注: 参照プロジェクトや複製するプロジェクト名にエラーがある間は、このボタンは有効になりません。

8. **はい** をクリックして複製処理を開始します。
9. プロジェクトが正しく複製されたことを示すダイアログが表示されます。**OK** をクリックします。**プロジェクト** リストに複製したプロジェクトが表示されます。
10. **OK** をクリックして **管理** ダイアログを閉じます。



注: 元のプロジェクトにある結果はコピーされません。

プロジェクトの削除

Silk Test Workbench データベースからプロジェクトを削除するには、管理者としてログインする必要があります。

1. **ツール** > **管理** を選択します。
または、**開始画面** の **タスク** ペインで **管理** をクリックします。
管理 ダイアログ ボックスが開きます。

2. **プロジェクト** タブをクリックします。
3. 削除するプロジェクトを選択します。
4. **削除** をクリックします。プロジェクトの削除を確認するメッセージ ボックスが表示されます。
5. **OK** をクリックして、プロジェクトを完全に削除します。


ユーザーの管理

管理者は、Silk Test Workbench のリソースや情報にアクセスすることを許可されたユーザーのリストを管理する必要があります。

データベースに接続中のユーザーの表示

Silk Test Workbench データベースに接続中の接続ユーザーを表示するには：

Silk Test Workbench メニューから、**ツール > 接続ユーザーの表示** を選択します。**接続ユーザー** ウィンドウが開き、すべてのユーザーがリスト表示されます。表示される列は、ユーザー名、ユーザーのフルネーム、データベースに接続したマシン名、データベースにログインした日時になります。このリストは、5 分ごとに更新されます。

 **注:** 同じユーザーが複数のマシンからデータベースにログインした場合や、同じマシン上で複数の Silk Test Workbench を実行している場合など、同じユーザーがリストに複数表示される場合があります。

ユーザーの追加

Silk Test Workbench データベースにユーザーを追加するには、管理者としてログインする必要があります。ユーザー プロファイルを編集すると、プロジェクトに対するユーザーのデフォルト権限をいつでも上書きできます。

1. **ツール > 管理** を選択します。
または、**開始画面** の **タスク** ペインで **管理** をクリックします。
管理 ダイアログ ボックスが開きます。
2. **ユーザー** タブをクリックします。
3. **追加** をクリックします。**ユーザーの追加** ダイアログ ボックスが開きます。
4. ユーザーの認証情報を Silk Test Workbench が保存できる Windows 認証をこのユーザーに対して使用したい場合は、**ドメイン** フィールドにドメインを入力します。Silk Test Workbench 認証を使用したい場合は、**ドメイン** フィールドを空のままにします。
5. **ユーザー ID** テキスト ボックスに、ユーザーのログオン ID を入力します。
6. **パスワード** テキスト ボックスに、ユーザーのパスワードを入力します。
7. **パスワードの確認入力** テキスト ボックスに、同じパスワードを入力します。
8. **フルネーム** テキスト ボックスに、ユーザーの名前を入力します。
9. **種類** リストを使用して、ユーザーが管理者であるかどうかを指定します。
10. **デフォルトの権限** リストを使用して、ユーザーに対して新しい Silk Test Workbench プロジェクトに対するデフォルトのアクセス レベルを割り当てます。
以下の種類の権限から選択します。
 - **アクセス不可**
ユーザーは、プロジェクト情報の表示または変更、あるいは記録および再生の実行ができません。
 - **読み取り専用**
ユーザーは、プロジェクト情報を表示できますが、資産を変更したり、記録および再生を実行したりできません。

- **実行者**

ユーザーは、プロジェクト情報を表示して結果を変更できますが、他の資産は変更できません。

- **スクリプト作成者**

ユーザーは、プロジェクト情報の表示と資産の変更を実行できます。

- **フルアクセス**

ユーザーは、すべての資産に対する完全な読み取り／書き込みアクセス権を持ちます。

11OK をクリックします。

ユーザーの削除

Silk Test Workbench データベースから他のユーザーを削除するには、管理者としてログインする必要があります。

1. **ツール** > **管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **ユーザー** タブをクリックします。

3. 削除するユーザーを選択します。



注: 管理権を持つ他のユーザーが存在する場合だけ、デフォルトのユーザー *Admin* を削除することができます。

4. **削除** をクリックします。選択したユーザーの削除を確認するメッセージ ボックスが表示されます。

5. **OK** をクリックして、ユーザーを完全に削除します。

ユーザー プロファイルの編集

1. **ツール** > **管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **ユーザー** タブをクリックします。

3. 変更するユーザー プロファイルを選択して、**変更** をクリックします。**ユーザーの変更** ダイアログ ボックスが開きます。

4. 適切なテキスト ボックスに新しい情報を入力します。

5. **OK** をクリックして、変更内容を保存します。

プロジェクトに対するユーザーのアクセス権の変更

1. **ツール** > **管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **ユーザー** タブをクリックします。

3. アクセス権限を変更するユーザーを選択します。

管理 ダイアログ ボックスの右側のパネルには、ユーザーがアクセスできるプロジェクト、および各プロジェクトに対するユーザーのアクセス レベルがリストされます。

4. プロジェクトの **ユーザー アクセス** リストを使用して、ユーザーに割り当てる新しいアクセス権限レベルを選択します。

5. **OK** をクリックします。

データベースからのユーザーのログアウト

管理者として資産のメンテナンス タスクを実行するために、資産メンテナンス モードをオンにすると、管理者以外のユーザーをデータベースからログアウトさせることができます。

1. 資産メンテナンス モードをオンにするには、Silk Test Workbench メニューから **ツール > 資産メンテナンス モード** を選択します。資産メンテナンス モードをオンにすると、次のようになります。
 - Silk Test Workbench ウィンドウの下部にあるステータス バーに、資産メンテナンス モードがアクティブであることが表示されます。
 - データベース メンテナンス モードを開始すると、管理者以外のすべてのユーザーがログアウトされます。管理者以外のユーザーが Silk Test Workbench ダイアログを開いており、ログアウトできない場合は、Silk Test Workbench にログインしたままになりますが、データベース中の資産を変更できなくなります。
 - すべての開いている資産は保存され、閉じられます。
 - 実行中の記録や再生は、可能な限り速やかに終了されます。
 - 資産メンテナンス モードがアクティブな間は、管理者以外のユーザーは Silk Test Workbench データベースにログインできません。
2. データベースが資産メンテナンス モード中に、管理者が Silk Test Workbench からログアウトしようとする、ログアウトする前に資産メンテナンス モードをオフにするかどうかを確認されます。**はい** をクリックすると、資産メンテナンス モードはオフになります。

グループの管理

Windows ドメイン アカウントのグループを同じ権限でデータベースにログインできるようにします。

グループが共有データベースを持つ場合、単にそのグループを追加して、すべてのグループ メンバが同じ権限でデータベースにログインすることができます。新しいメンバがグループに追加されると、データベースへのアクセス権が自動的に与えられ、メンバがグループから削除されると、データベースへのアクセス権がなくなります。

ローカルマシンのすべてのユーザーがデータベースにアクセスできるようにするには、**BUILTIN¥ALL Users** をグループ リストに追加します。新しい Access データベースには、デフォルトでこのグループがグループ リストが追加され、新しいユーザーが簡単に開始できるようになっています。

グループの追加

同じ資格情報でデータベースにアクセスするために、Silk Test Workbench データベースを共有するグループを有効化するには、グループ プロファイルを追加します。

1. **ツール > 管理** を選択します。
または、**開始画面** の **タスク** ペインで **管理** をクリックします。
管理 ダイアログ ボックスが開きます。
2. **グループ** タブをクリックします。
3. **追加** をクリックします。**グループの追加** ダイアログ ボックスが開きます。
4. グループの認証情報を Silk Test Workbench が保存できる Windows 認証をこのグループに対して使用したい場合は、**ドメイン** フィールドにドメインを入力します。Silk Test Workbench 認証を使用したい場合は、**ドメイン** フィールドを空のままにします。
5. **種類** リストを使用して、グループが管理者グループであるかどうかを指定します。
6. **デフォルトの権限** リストを使用して、グループに対して新しい Silk Test Workbench プロジェクトに対するデフォルトのアクセス レベルを割り当てます。
以下の種類の権限から選択します。

- **アクセス不可**

ユーザーは、プロジェクト情報の表示または変更、あるいは記録および再生の実行ができません。

- **読み取り専用**

ユーザーは、プロジェクト情報を表示できますが、資産を変更したり、記録および再生を実行したりできません。

- **実行者**

ユーザーは、プロジェクト情報を表示して結果を変更できますが、他の資産は変更できません。

- **スクリプト作成者**

ユーザーは、プロジェクト情報の表示と資産の変更を実行できます。

- **フルアクセス**

ユーザーは、すべての資産に対する完全な読み取り/書き込みアクセス権を持ちます。

7. **OK** をクリックします。

グループの削除

Silk Test Workbench データベースからグループを削除するには、管理者としてログインする必要があります。

1. **ツール** > **管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **グループ** タブをクリックします。

3. 削除するグループを選択します。

4. **削除** をクリックします。選択したグループの削除を確認するメッセージ ボックスが表示されます。

5. **OK** をクリックして、グループを完全に削除します。

グループの変更

1. **ツール** > **管理** を選択します。

または、**開始画面** の **タスク** ペインで **管理** をクリックします。

管理 ダイアログ ボックスが開きます。

2. **グループ** タブをクリックします。

3. 変更したいグループを選択し、**変更** をクリックします。**グループの変更** ダイアログ ボックスが開きます。

4. 適切なテキスト ボックスに新しい情報を入力します。

5. ユーザーの種類と権限を適切なリスト ボックスから選択します。

6. **OK** をクリックして、変更内容を保存します。

UI トピック

Silk Test Workbench のデスクトップは、テストにかかわるすべての操作の開始点になります。メニューバーを除き、デスクトップに表示されるウィンドウやペインをはじめとする各種機能の表示と外観を制御することができます。

開始画面

開始画面 は、Silk Test Workbench の機能を使い始める拠点であり、アプリケーションのテスト ソリューションをすばやく作成し始めることができます。

アプリケーションのテストをすばやく、最低限の手順で開始できるようにするため、**開始画面** には次の領域が含まれています。

領域 説明

開始 ビジュアル テスト、.NET スクリプト、キーワード駆動テストをワンクリックで記録したり、開いたりできます。

資産 最近使ったビジュアル テスト、.NET スクリプト、キーワード駆動テストをワンクリックで開くことができます。最近使った資産のリストは、リアルタイムで更新されるので、変更や更新により、現在の資産がリストの先頭に移動します。資産の右側にあるアイコンをクリックすることによって、資産を再生したり、資産の最後の実行結果を表示したりすることができます。



注: **開始画面** 設定 (**ツール > オプション > 開始画面**) では、最近使った資産のリストに表示する項目の数を制御できます。

新機能 Silk Test Workbench の現在のバージョンに含まれている新しい機能や機能強化を一覧します。Silk Test Workbench の更新が利用可能な場合も、ここに表示されます。



注: Silk Test Workbench の起動時に開始画面を表示しないように選択している場合、**ヘルプ > 製品更新の確認** をクリックして利用可能な更新を確認できます。

入門ガイド 生産性向上のため、広く使われるタスクや有用なヘルプ トピックへの直接リンクを提供します。

リソース サポート技術情報や Silk Test コミュニティなどの Silk Test Workbench に関連した情報へのリンクを提供します。

お知らせとコミュニティ Silk Test コミュニティの最新のフォーラム記事とブログ投稿へのリンクを提供します。

フラグ ビジュアル テストのテスト ステップ、およびフラグが付けられてフォローアップ対象として割り当てられた結果が表示されます。**フラグ** ペインを使用すると、テストスイートやテストプロジェクトについての情報を交換でき、データベースのテストプロジェクトのほかのテスト担当者やユーザーと共同作業が行えます。

デフォルトでは、フラグは作成日および表示される資産で分類されます。フラグの情報をすばやく表示するには、マウスのポインタをフラグの上に移動します。フラグの説明、更新データ、プロジェクト、フラグが割り当てられているユーザーの ID を示すツール ヒントが表示されます。**開始画面のフラグ** オプション (**ツール > オプション > 開始画面 > フラグ**) を変更して、フラグの表示方法を変更できます。

領域 説明

フラグに関連付けられているビジュアル テストまたは結果を開くには、**フラグ** ペインでフラグをダブルクリックします。



注: **フラグ** ペインは、少なくとも 1 つのフラグがログインしているユーザーに割り当てられているか、少なくとも 1 つのフラグがフィルタで表示されるように設定されている場合にのみ、**開始画面** に表示されます。

開始画面 の右上隅にあるウィンドウの **閉じる** をクリックすると、**開始画面** を閉じることができます。



ヒント: ツールバーの **開始画面** をクリックすることで、メイン画面から **開始画面** にいつでもアクセスできます。また、いつでも Ctrl+Alt+S を押すか、または **表示** > **開始画面** を選択して、開始画面を表示できます。



注: ユーザー インターフェイスは、Windows 標準の大きいフォントや小さいフォントを使用します。カスタムのフォント サイズを使用すると、ユーザー インターフェイスのテキスト表示が不正確になる可能性があります。

Silk Test Workbench のメニュー

このセクションでは Silk Test Workbench メニューについて説明します。

メニュー コマンドは、現在使用しているウィンドウ、資産、またはテスト モードの種類に応じて、有効または無効にできます。

[ファイル] メニュー

ファイル メニューには以下のコマンドがあります。

コマンド (ショートカット)	説明
新規作成 (Ctrl+N)	新規作成 ウィンドウを表示し、任意の種類の資産を作成することができます。
開く (Ctrl+O)	アセットブラウザ を開き、資産を開くことができます。前回作業した資産の種類が強調表示されます。
閉じる	現在フォーカスが設定されているデスクトップのウィンドウを閉じます。
新しいバージョンとして保存 (Ctrl+S)	<ul style="list-style-type: none">資産の現在の状態を資産の新しいバージョンとして保存し、増分の数値識別子をそのバージョンに割り当てます。ショートカット キーの組み合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
現在のバージョンとして保存 (Ctrl+S)	資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「 資産保存時のデフォルト動作を設定する 」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
名前を付けて保存	名前を付けて保存 ダイアログ ボックスを開いて、現在の資産を別の名前で保存します。
すべて保存 (Ctrl+Shift+L)	<ul style="list-style-type: none">開いているすべての資産の現在の状態を資産の新しいバージョンとして保存し、各資産のバージョン識別子をインクリメントします。ショートカット キーの組み合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合

コマンド (ショートカット)	説明
	<p>に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。</p> <ul style="list-style-type: none"> 開いているすべての資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
複製	アセット ブラウザ で資産を選択してからこのメニュー項目を選択し、新しい名前を使って資産を複製します。
名前の変更	アセット ブラウザ で資産を選択してからこのメニュー項目を選択し、選択した資産の名前を変更します。
削除 (Del)	選択した資産を削除します。
すべてのバージョンを表示	アセット ブラウザ で資産を選択してからこのメニュー項目を選択し、資産のすべてのバージョンを表示します。
要約情報 (Alt+Enter)	ビジュアル テストまたはスクリプトから 要約情報 ダイアログ ボックスを表示します。
ページ設定	ブラウザ コントロールの ページ設定 を表示します。このダイアログ ボックスでの設定は、現在選択されている資産に適用され、[印刷プレビュー] ダイアログに表示されません。
印刷プレビュー	印刷プレビュー ダイアログ ボックスを表示して、現在の資産を印刷する前にそのレイアウトを確認できます。
印刷 (Ctrl+P)	[印刷] ダイアログ ボックスを表示して、選択した資産を印刷できます。
資産のエクスポート	資産のデータベース間でのエクスポートは、 資産エクスポート ウィザードを使用して行います。詳細については、「 資産のエクスポート 」を参照してください。
資産のインポート	資産のデータベース間でのインポートは、 資産インポート ウィザードを使用して行います。詳細については、「 資産のインポート 」を参照してください。
最近使った資産	最近アクセスしたテスト資産のサブメニューを表示します。
終了	Silk Test Workbench を終了します。

[編集] メニュー

編集 メニューには以下のコマンドがあります。

コマンド	説明
元に戻す (Ctrl+Z)	直前の編集を元に戻します。資産の比較、またはマージ時には利用できません。
やり直し (Ctrl+Y)	直前の 元に戻す 操作をやり直します。資産の比較、またはマージ時には利用できません。
切り取り (Ctrl+X)	選択した項目を切り取って、クリップボードに貼り付けます。
コピー (Ctrl+C)	選択した項目をコピーします。
貼り付け (Ctrl+V)	選択した項目を貼り付けます。
すべて選択 (Ctrl+A)	現在のビジュアル テストのすべてのテスト ステップ、または現在のテスト スクリプトのコード行を選択します。

コマンド	説明
検索と置換	スクリプト内の特定のコードを検索し、必要に応じて置換します。
指定行へ移動 (Ctrl+G)	スクリプト内の特定の行へ移動します。
ブックマーク	ブックマークのオンとオフの切り替え、前のブックマークまたは次のブックマークへの移動、またはスクリプト内のブックマークのクリアを実行します。
行コメント (Ctrl+K)	スクリプトで現在選択されている行 (複数) をコメントアウトして無効化します。コード ブロック全体をコメントアウトするために、このショートカットを使用できます。また、 Ctrl+F11 を使用することもできます。
行コメント解除 (Ctrl+Shift+K)	スクリプトで現在選択されている行 (複数) のコメントを解除して有効化します。
フラグ	現在選択されているフラグを編集または削除します。
画面の更新	選択したテスト ステップについてテスト アプリケーションから画面情報を更新します。テスト アプリケーションの画面が実行されており、更新が必要なステップが選択されている必要があります。
条件	条件 ダイアログ ボックスを表示して、検証全体の合格基準とする、再生での検証の合格の割合を設定できます。
フィルタの管理	結果に表示するデータをフィルタ処理するための フィルタの管理 ダイアログ ボックスを表示します。フィルタは、現在表示されているビューにかかわらず、 要約 ビューと 詳細 ビューの両方に適用されます。

[操作] メニュー

操作 メニューには以下のコマンドがあります。

コマンド	説明
記録	すべてのユーザー操作を、アクティブなビジュアル テストまたはテスト スクリプトの現在の位置に記録します。
一時停止	現在再生中のビジュアル テストまたはテスト スクリプトを一時停止して、実行が中断された地点を強調表示します。
再生 (F5)	アクティブなビジュアル テストまたはスクリプトを再生します。
停止	現在アクティブなビジュアル テストまたはスクリプトの再生を中止します。
基本状態の実行	テスト対象アプリケーションを基本状態に戻します。基本状態とは、テストの実行を始める前にアプリケーションがあるべき既知の安定した状態を言います。詳細については、「 基本状態 」を参照してください。
相違点の表示	選択した 2 つの資産の違いを比較します。
画面の更新	選択したテスト ステップに対して、テスト アプリケーションから画面情報を更新します。テスト アプリケーションの画面が実行されており、更新が必要なステップが選択されている必要があります。

[表示] メニュー

表示 メニューには以下のコマンドがあります。

コマンド	説明
移動	<p>以下に挙げる 1 つまたは複数の項目に移動するためのコマンドを示すサブメニューを表示します。</p> <p>[詳細] ページ : 結果の 合格 タブまたは 失敗 タブで選択したステップから、結果の 詳細 タブの同じステップに移動します。</p> <p>ビジュアルテストのウィンドウ : 結果ウィンドウからビジュアルテストウィンドウに移動します。</p> <p>結果ウィンドウ : ビジュアルテストウィンドウから結果ウィンドウに移動します。</p>
ロジック ツールボックス	<p>テスト ロジック デザイナ へのリンクを表示する [テスト ロジック ツールボックス] を表示します。</p>
新しいウィンドウで開く	<p>選択したイメージを 画面プレビュー に表示するか、または選択した結果を新しいウィンドウに表示します。</p>
テスト ステップ	<p>ビジュアル ナビゲータ の テスト ステップ ウィンドウまたは 結果 ウィンドウで、特定のステップの表示を切り替えます。</p> <p>ステップのみ : テスト アプリケーションに対してテスト操作を実行するステップを表示します。</p> <p>画面のみ : テストに関連付けられた画面を表示するステップを表示します。</p> <p>ステップと画面 : 操作を実行するステップと画面を表示するステップの両方を表示します。</p> <p>ステップの説明 : 表示されるテスト ステップに入力されている説明を表示します。</p>
列	<p>結果 ウィンドウの 詳細 タブに結果情報を表示します。</p> <p>基本表示 : 標準的な テスト ステップ ペインの情報、および 結果 と 結果の詳細 の追加の列を表示します。</p> <p>詳細表示 : 各ステップの詳細情報を表示します。</p>
開始画面 (Ctrl+Alt+S)	<p>開始画面 を表示します。</p>
アセット ブラウザ (Ctrl+Alt+A)	<p>アセット ブラウザ を開きます。</p>
エラー リスト	<p>テスト ステップ ペインで、エラーを赤色で表示します。</p>
出力	<p>出力 ウィンドウを表示します。</p>
インクルード	<p>インクルード ウィンドウを表示します。</p>
キーワード	<p>作業中のプロジェクトのすべてのキーワード、および統合した Silk Central インスタンスで定義されたキーワードを一覧する キーワード エディタ を表示します。</p>
更新	<p>ウィンドウを更新します。</p>
ズーム	<p>画面プレビュー の現在のイメージのサイズを拡大/縮小します。アプリケーション ウィンドウがアクティブな場合に有効になります。</p>
ツール バー	<p>ツール バーの表示/非表示を切り替えます。</p>
ステータス バー	<p>現在のセッション情報を表示するステータス バーを表示します。</p>
レイアウトの復元	<p>ビジュアル ナビゲータ のレイアウトを元の表示設定に戻します。</p>

[挿入] メニュー

挿入 メニューには以下のコマンドがあります。

コマンド	説明
テスト ロジック	現在のビジュアル テストに対してロジックを作成します。詳細については、「 ツールバー 」または「 テスト ロジック 」を参照してください。
コントロール	次のひとつからビジュアル テストにコントロールを挿入します。 <ul style="list-style-type: none">• テスト対象アプリケーション。• 画面プレビュー。• オブジェクトの識別 ダイアログ。
ファイル	外部ファイルをスクリプトに挿入します。スクリプトがアクティブな場合に有効になります。
ビジュアル テスト	ビジュアル テストの参照 ダイアログ ボックスを開いて、ビジュアル テストをアクティブなビジュアル テストに追加します。ビジュアル テストがアクティブな場合に有効になります。
.NET スクリプト	.NET スクリプトの参照 ダイアログ ボックスを開いて、.NET スクリプトをアクティブなビジュアル テストに追加します。ビジュアル テストがアクティブな場合に有効になります。
外部プログラム	アクティブなビジュアル テストの再生中に、選択したポイントでアプリケーションまたはプログラムを開始します。
式	式デザイナー を表示して、アクティブなビジュアル テストに式を挿入できます。
検証資産	ビジュアル テストの現在のステップに既存の検証資産を挿入します。現時点では、イメージ検証のみが検証資産としてサポートされています。詳細については、「 イメージ検証 」を参照してください。
結果コメント	ビジュアル テストの現在のステップに結果のコメントを挿入します。詳細については、「 ビジュアル テストへのコメントの挿入 」を参照してください。
ラベル	空白の編集可能なステップを、プロパティ値を指定せずにビジュアル テストに挿入します。
エラー処理	ビジュアル テストの現在のステップにエラー処理を挿入します。
メッセージ ボックス	メッセージ ボックスのテスト ステップを挿入します。
コメント	ビジュアル テストの現在のステップにコメントを挿入します。
再生設定	再生設定とその値を現在のテストに設定するか、または現在のテストから取得します。
コントロールのプロパティ	空のステップを挿入し、そのステップを使用してテスト アプリケーションのコントロールからプロパティをキャプチャして、変数に設定します。ステップの プロパティ ペインで、 ロケータ フィールドの 識別 ボタンを使用してコントロールをキャプチャし、コントロールの特定のプロパティを選択します。
プログラム フロー	テスト フローを制御するステップを挿入するか、またはテストの信頼性を高めます。
同期とタイミング	オブジェクトを待機するテスト ステップ、指定した時間だけビジュアル テストの再生を遅らせるテスト ステップ、または再生のタイマーを作成するテスト ステップを挿入します。

コマンド	説明
SAP eCATT	<ul style="list-style-type: none"> 引数の取得 引数の設定 <p>ビジュアルテストと SAP eCATT 資産の間で変数を受け渡しできるようにします。SAP が Silk Test Workbench と同じコンピュータにインストールされている場合に有効になります。詳細については、「SAP eCATT の操作」を参照してください。</p>
アクティブ データ	<p>次の操作があります。</p> <p>新規作成 新しいアクティブ データ資産を作成して、ビジュアルテストに関連付けます。詳細については、「アクティブ データ資産のビジュアルテストへの関連付け」を参照してください。</p> <p>既存の関連付け 既存のアクティブ データ資産をビジュアルテストに関連付けます。詳細については、「アクティブ データ資産のビジュアルテストへの関連付け」を参照してください。</p> <p>直ちに保存 アクティブ データ ファイルを直ちに保存するステップをビジュアルテストに挿入します。詳細については、「ビジュアルテスト内で更新されたアクティブ データの即時保存」を参照してください。</p> <p>保存のキャンセル アクティブ データ ファイルを保存する前の要求をキャンセルします。</p>
変数	<p>ビジュアルテストに対して、ローカル変数の追加や、グローバルパラメータの設定または取得を行います。詳細については、「ビジュアルテストにおける変数」を参照してください。</p>
フラグ	<p>選択したテスト ステップにフラグまたはメモを設定します。詳細については、「フラグ」を参照してください。</p>
画面の更新	<p>選択したテスト ステップに対して、テスト アプリケーションから画面情報を更新します。テスト アプリケーションの画面が実行されており、更新が必要なステップが選択されている必要があります。</p>

[デバッグ] メニュー

テストのデバッグの詳細については、「[テストのデバッグ](#)」を参照してください。**デバッグ** メニューには以下のコマンドがあります。

コマンド (ショートカット)	使用可能な資産	説明
コンパイル (F7)	<ul style="list-style-type: none"> スクリプト 	スクリプトの構文がコンパイルのすべての条件を満たしているかをチェックし、コードをコンパイルし、エラーが発生した場合は表示します。
ステップイン (F8)	<ul style="list-style-type: none"> スクリプト ビジュアルテスト 	ビジュアル ステップの場合、テスト ステップを一度に 1 つずつ再生します。スクリプトの場合、ステートメントを一度に 1 つずつ再生します。
ステップ オーバー (Shift+F8)	<ul style="list-style-type: none"> スクリプト ビジュアルテスト 	埋め込まれているビジュアルテストまたはスクリプト全体を再生し、元のビジュアルテストまたはスクリプト内の次のステップで停止してデバッグモードになります。VB .NET スクリプト

コマンド (ショートカット)	使用可能な資産	説明
ステップ アウト (Ctrl+Shift+F8)	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	<p>プドで押下された場合は、関数呼び出しのステップ オーバーも行います。</p> <p>別のビジュアル テストから再生しているビジュアル テスト内の残りのステップをすべて実行したあと、元のビジュアル テスト内の次のステップで再生が一時停止します。スクリプトの場合は、プロシージャ内の残りすべてのコードを 1 つのステートメントのように実行し、そのプロシージャを最初に呼び出したプロシージャ内の次のステートメントに移ります。</p>
カーソルまで実行 (Ctrl+F8)	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	現在選択されているテスト ステップまでビジュアル テストを再生するか、または現在選択されているステートメントまでスクリプトを再生します。
カーソルから実行	<ul style="list-style-type: none"> ビジュアル テスト 	現在選択されているテスト ステップからビジュアル テストを再生します。
ブレイクポイントの設定/クリア (F9)	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	選択したステップまたはコード行にブレイクポイントを設定するか、または選択したステップまたはコード行に設定されているブレイクポイントをクリアします。
すべてのブレイクポイントのクリア (Ctrl+Shift+F9)	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	すべての開いているビジュアル テストと VB .NET スクリプトのすべてのブレイクポイントをクリアします。
実行ポイント/次のステートメントの設定 (Ctrl+F9)	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	次のステートメントまたはポイントまで再生を実行します。デバッグ モードの場合に有効になります。
ローカル変数	<ul style="list-style-type: none"> スクリプト ビジュアル テスト 	現在のビジュアル テストまたはスクリプト、および埋め込まれているビジュアル テストまたはスクリプトについて、すべての変数とその値をリストするウィンドウを表示します。

[ツール] メニュー

ツール メニューには以下のコマンドがあります。

コマンド	説明
データベースの変更	ログオン ダイアログ ボックスを表示して、別のデータベースにログオンします。
パスワードの変更	パスワードの変更 ダイアログ ボックスを表示して、現在のユーザーのログオン パスワードを変更します。
オブジェクトの識別 (Ctrl+Shift+I)	選択したオブジェクトのロケータを記録できる オブジェクトの識別 ダイアログ ボックスを開きます。
管理	ユーザーを管理し、プロジェクトにアクセスします。
オプション	構成設定を表示し、それらの設定をカスタマイズできます。
キーワード ライブラリのアップロード	キーワード ライブラリを Silk Central にアップロードします。Silk Test Workbench が Silk Central と統合され

コマンド	説明
カスタム コントロールの管理	<p>ている場合にのみ有効になります。詳細については、「Silk Central へのキーワード ライブラリのアップロード」を参照してください。</p> <p>カスタム コントロールの管理 ダイアログ ボックスを表示します。</p>
アプリケーション構成の編集	<p>アプリケーション構成の編集 ダイアログ ボックスを表示して、特定のプロジェクトでキーワード駆動テストによってテストされるアプリケーションを指定したり、アプリケーションの起動方法を設定することができます。詳細については、「アプリケーション構成の編集」を参照してください。</p>
リモート ロケーションの編集	<p>リモート ロケーション ダイアログ ボックスを表示して、ブラウザやデバイスにアクセスするリモート ロケーションを指定することができます。詳細については、「リモート ロケーションの編集」を参照してください。</p>
ツールのカスタマイズ	<p>ツール メニューにメニュー項目を追加します。</p>
SAP eCATT	<p>SAP eCATT で作業するためのコマンドにアクセスできます。Silk Test Workbench が SAP eCATT と統合されている場合にのみ有効になります。詳細については、「SAP eCATT の操作」を参照してください。</p>

[ツール] メニューのカスタマイズ

Silk Test Workbench では、**ツール** メニューにメニュー項目を追加できます。これらのメニュー項目により、頻繁に使用するプログラムを簡単に開くことができるようになります。メニュー項目の追加は、頻繁に使用するプログラムやドキュメントにすばやくアクセスする方法を提供するという点で、ショートカットの使用に似ています。たとえば、メモ帳用のカスタム メニュー項目を作成すると、Silk Test Workbench での作業を中断せずに、**ツール** メニューからメモ帳を起動できるようになります。

新しいコマンドに関する情報を入力したあとは **追加** をクリックしないでください。クリックすると、別の空白エントリが (**新しいツール**) として追加され、**ツール** メニューの項目として表示されてしまいます。

- ツール** > **ツールのカスタマイズ** をクリックします。 **カスタマイズ** ダイアログ ボックスが開きます。
- 新しい項目を作成するには、**追加** をクリックします。 **メニュー内容** リストに空白エントリが (**新しいツール**) として表示されます。 (**新しいツール**) エントリは、**メニュー テキスト** テキスト ボックスにも表示されます。
- メニュー テキスト** テキスト ボックスで、メニュー項目の名前を入力します。
この名前は、**ツール** メニューに表示されるようになります。たとえば、このメニュー項目によってメモ帳を起動する場合は、「メモ帳」と入力します。
- コマンド** テキスト ボックスで、**参照** をクリックして、メニュー項目を実行するファイルに移動して選択します。
ファイルを選択するには、**プログラムの選択** ダイアログ ボックスでそれを強調表示して、**開く** をクリックします。
ファイルのパスと名前がわかっている場合は、**コマンド** テキスト ボックスにそれを入力することもできます。
- 引数** テキスト ボックスに、**コマンド** テキスト ボックスのファイルを実行するために必要なパラメータを入力します。
コマンドが必要ない場合は、このテキスト ボックスを空のままにします。
- 初期ディレクトリ** テキスト ボックスに、元の項目を含むパスとディレクトリ、またはメニュー項目によって実行されるファイルに必要なすべての関連ファイルを入力します。

プログラムによっては、複数の場所のファイルを使用する必要があります。メニュー項目によって実行されるファイルがこれらのファイルを見つけられるように、これらのファイルが置かれているフォルダを指定する必要があります。

7. **OK** をクリックします。**カスタマイズ** ダイアログ ボックスが閉じ、項目が **ツール** メニューに追加されます。

[ウィンドウ] メニュー

ウィンドウ メニューには以下のコマンドがあります。

コマンド	説明
上下に並べて表示	開いているすべてのウィンドウを上下に並べて表示します。
左右に並べて表示	開いているすべてのウィンドウを左右に並べて表示します。
重ねて表示	開いているすべてのウィンドウを重ねて表示します。
アイコンの整列	最小化したスクリプト、ビジュアルテストなどをウィンドウの最下部に整列します。
開いているファイルのリスト	開いているビジュアルテスト、結果、スクリプト、またはウィンドウのサブメニューを表示します (アセットブラウザ や 開始画面 など)。

[ヘルプ] メニュー

ヘルプ メニューには以下のコマンドがあります。

コマンド	説明
目次	ヘルプの目次を表示します。
検索	ヘルプの 検索 タブを開きます。
索引	ヘルプの 索引 タブを開きます。
チュートリアル	Silk Test Workbench のチュートリアルを開きます。
製品更新の確認	Silk Test Workbench の新しいバージョンが利用可能かどうかを確認します。
Silk Test Workbench のバージョン情報	Silk Test Workbench の製品情報を表示します。

ツールバー

メニューの **表示** > **ツールバー** を開き、ツールバーの左側にあるチェック ボックスのオン/オフを切り替えて、ツールバーを表示/非表示します。Silk Test Workbench では、以下のツールバーを利用できます。

ファイル 資産管理を行うための以下のアクションが提供されます。

ボタン	説明
新規作成 (Ctrl+N)	新規作成 ウィンドウを表示し、任意の種類の資産を作成することができます。
開く (Ctrl+O)	アセットブラウザ を開き、資産を開くことができます。前回作業した資産の種類が強調表示されます。
新しいバージョンとして保存 (Ctrl+S)	<ul style="list-style-type: none">資産の現在の状態を資産の新しいバージョンとして保存し、増分の数値識別子をそのバージョンに割り当てます。ショートカットキーの組み

ボタン	説明
	<p>合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。</p> <ul style="list-style-type: none"> 資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
すべて保存 (Ctrl+Shift+L)	<ul style="list-style-type: none"> 開いているすべての資産の現在の状態を資産の新しいバージョンとして保存し、各資産のバージョン識別子をインクリメントします。ショートカット キーの組み合わせは、デフォルトの保存動作が 新しいバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。 開いているすべての資産の現在の状態を資産の現在のバージョンとして保存します。ショートカット キーの組み合わせは、デフォルトの保存動作が 現在のバージョンとして保存 に設定されている場合に有効です。詳細については、「資産保存時のデフォルト動作を設定する」を参照してください。このコマンドは、資産を最初に保存するときは無効です。
印刷 (Ctrl+P)	[印刷] ダイアログ ボックスを表示して、選択した資産を印刷できます。
印刷プレビュー	印刷プレビュー ダイアログ ボックスを表示して、現在の資産を印刷する前にそのレイアウトを確認できます。
索引	ヘルプの 索引 タブを開きます。

編集 編集を行うための以下のアクションが提供されます。

ボタン	説明
切り取り (Ctrl+X)	選択した項目を切り取って、クリップボードに貼り付けます。
コピー (Ctrl+C)	選択した項目をコピーします。
貼り付け (Ctrl+V)	選択した項目を貼り付けます。
元に戻す (Ctrl+Z)	直前の編集を元に戻します。資産の比較、またはマージ時には利用できません。
やり直し (Ctrl+Y)	直前の 元に戻す 操作をやり直します。資産の比較、またはマージ時には利用できません。

ボタン	説明
行コメント (<u>Ctrl+K</u>)	スクリプトで現在選択されている行 (複数) をコメントアウトして無効化します。コードブロック全体をコメントアウトするために、このショートカットを使用できます。また、 Ctrl+F11 を使用することもできます。
行コメント解除 (<u>Ctrl+Shift+K</u>)	スクリプトで現在選択されている行 (複数) のコメントを解除して有効化します。

アクション

記録操作を行うための以下のアクションが提供されます。

ボタン	説明
記録	すべてのユーザー操作を、アクティブなビジュアルテストまたはテストスクリプトの現在の位置に記録します。
再生 (F5)	アクティブなビジュアルテストまたはスクリプトを再生します。
一時停止	現在再生中のビジュアルテストまたはテストスクリプトを一時停止して、実行が中断された地点を強調表示します。
停止	現在アクティブなビジュアルテストまたはスクリプトの再生を中止します。
オブジェクトの識別 (<u>Ctrl+Shift+I</u>)	選択したオブジェクトのロケータを記録できる オブジェクトの識別 ダイアログボックスを開きます。

表示 テストロジック

アセットブラウザ や **開始画面** を表示します。

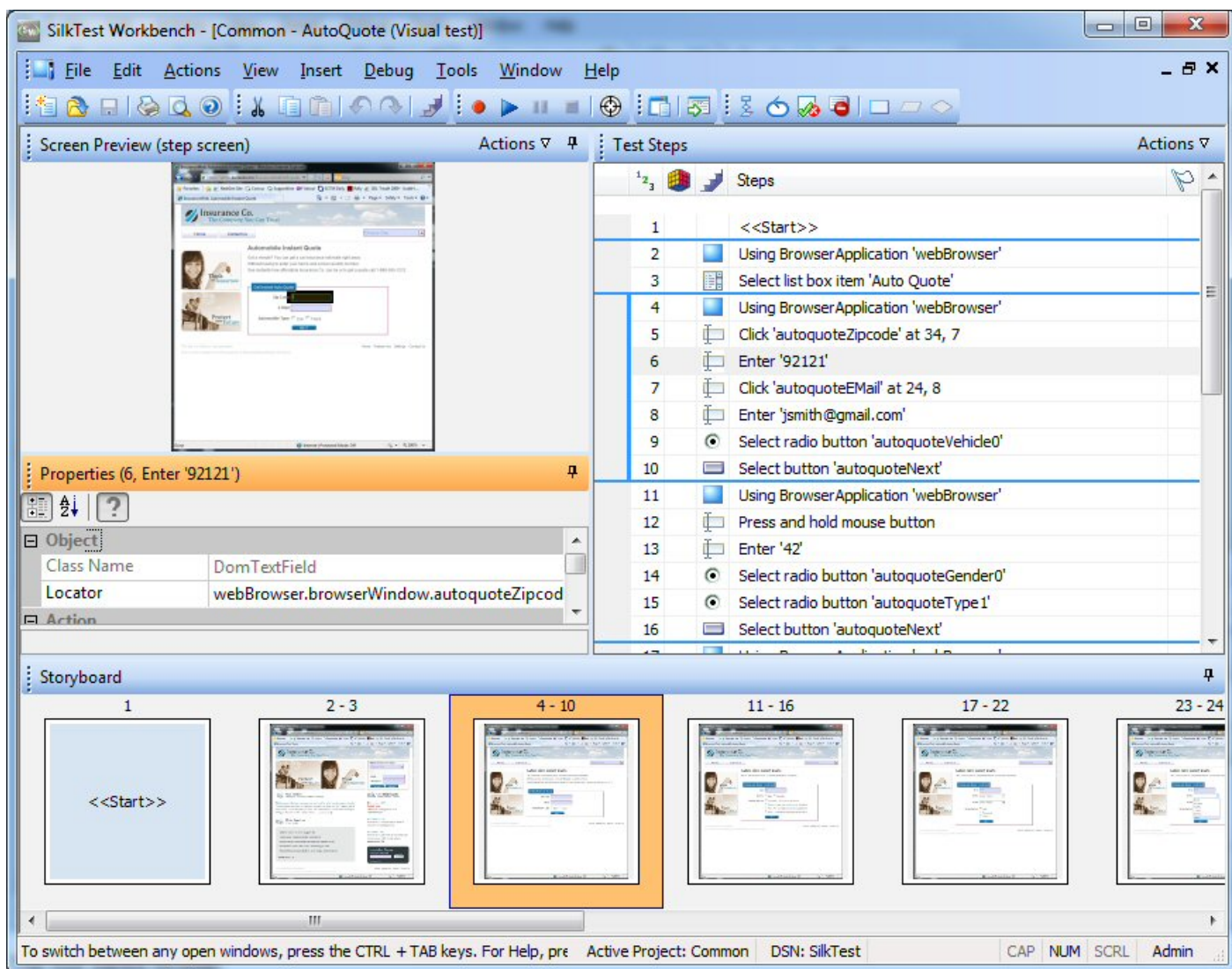
ビジュアルテストにロジックを追加するための以下のアクションが提供されます。

ボタン	説明
判断タイプのロジックの作成	判断ロジックでは、ユーザー定義の条件を評価して、その条件の戻り値に応じて一連のテストステップを実行します。
繰り返しタイプのロジックの作成	繰り返しロジックは、指定した回数、またはユーザー定義の条件が true になるまで、一連のテストステップを繰り返します。
検証タイプのロジックの作成 (<u>Ctrl+Alt</u>)	検証ロジックは、ユーザー定義の条件を評価して、合格/失敗メッセージをビジュアルテストまたはスクリプトの結果に送信します。
再生エラー処理ロジックの作成	エラー処理ロジックを設定すると、再生エラーをデバッグする必要がなくなります。エラー処理を設定するには、ビジュアルテストでエラーの検出ステップとその処理方法を設定します。
If タイプのロジック項目の挿入	作業中のビジュアルテストに条件ロジックを追加します。選択中のテストステップが、If と End If テストステップで囲まれます。テストステップは、条件に合致した場合にのみ実行されます。

ボタン	説明
Else If タイプのロジック項目の挿入	ビジュアルテストの If または Else If テストステップと End If テストステップ間でのみ使用できます。先行する If または Else If テストステップの条件が合致せず、この Else If テストステップの条件に合致した場合に、Else If と End If の間のテストステップが実行されます。
Else タイプのロジック項目の挿入	ビジュアルテストの If または Else If テストステップと End If テストステップ間でのみ使用できます。先行する If または Else If テストステップの条件が合致しない場合に、Else と End If の間のテストステップが実行されます。

ビジュアルナビゲータ

ビジュアルナビゲータは、ビジュアルテストの要素をグラフィカルに表すものであり、ポイントアンドクリックインターフェイスを通して各要素を操作できます。**ビジュアルナビゲータ**では、ビジュアルテストに関する情報が4つのペインに表示されます。4つのペイン全体でビジュアルテストの各ステップを総合的に把握できます。



4つのペインは以下のとおりです。

テストステップ ビジュアルテストの各ステップが技術的ではない明確な言語で示されます。

画面プレビュー ビジュアルテストの再生中に実行されるステップに応じてテスト対象アプリケーションのスナップショットが表示されます。


プロパティ ビジュアルテストのステップのプロパティが表示されます。

ストーリーボード サムネイルイメージを使用してビジュアルテストのフローが表示されます。サムネイルは、ビジュアルテストのステップの論理グループを表します。

 **注:**

Silk Test Workbench では、スナップショットは以下のような状況のときに取得されます。

- 記録中のすべての自動テストステップの前。

 **注:** SAP アプリケーションの場合、スナップショットはすべての自動テストステップの前ではなく、画面が変わったときに発生します。

- ビジュアルテストで Using ステップを実行した結果のスナップショット。

- 再生エラーが発生した場合。

画面プレビュー、**ストーリーボード**、および**プロパティ** ペインは **テスト ステップ** ペインと同期され、**テスト ステップ** ペインで選択したステップに関する情報が表示されます。このように、**テスト ステップ** ペインでステップを選択し、他のペインでそのステップに関する情報を確認することで、ステップのすべての側面を簡単に把握できます。

ビジュアル ナビゲータ では、ビジュアル テストを表示するだけでなく、**画面プレビュー** ペインと**プロパティ** ペインを使用して既存のビジュアル テストを拡張または更新することもできます。たとえば、ビジュアル テストを記録したあと、**プロパティ** ペインで、記録したプロパティのリテラル値を変数に置き換えることができます。また、テスト対象アプリケーションに変更が発生したときにビジュアル テストをすばやく更新する場合は、**画面プレビュー** の **画面の更新** 機能を使用して、以前にキャプチャした画面を更新できます。

ビジュアル ナビゲータ では、ビジュアル テストに使用すると同じペインを使用して、ビジュアル テストの再生結果も表示します。結果の場合、これらのペインには機能が付加され、**結果** ウィンドウ内に表示されます。ウィンドウにはツールバー オプションと複数のタブが表示されます。各タブには結果のさまざまな内容が表示されます。結果に固有の追加機能として、たとえば、**テスト ステップ** ペインに各ステップのステータス（合格または失敗）を表示する機能があります。さらに、**画面プレビュー** では、記録中にキャプチャされた画面と再生中にキャプチャされた画面の違いを比較し、テスト アプリケーションにアクセスすることなく、既存のビジュアル テストを更新できます。

[ビジュアル ナビゲータ] : [テスト ステップ] ペイン

テスト ステップ ペインは、ビジュアル テスト、およびビジュアル テストとスクリプトの結果を表示したり、管理したりするための中央の領域です。**テスト ステップ** ペインには、ビジュアル テストを構成する、記録または挿入された操作が 1 行ずつ表示されます。各操作はステップと呼ばれる個別の行で表されます。ステップは、テスト アプリケーションをテストするために行われる操作を、技術的でない明確な言語で表します。各ステップは、テスト アプリケーションに対して実行される操作またはテストを管理するために使用される操作に対応しています。

テスト アプリケーションと対話するステップは、**画面プレビュー** で、対応するアプリケーション画面要素のイメージによって表されます。また、各ステップは **ストーリーボード** のサムネイルに対応するようにグループ化されています。

各ステップには、ステップの操作に関する詳細を決定するプロパティが含まれています。ステップのプロパティは **プロパティ** ペインに表示されます。


ビジュアル テストの **テスト ステップ** ペインから、記録したビジュアル テストを確認し、以下の操作を追加して、テストの信頼性、柔軟性を向上させたり、検証、機能、およびフローを改良したりすることができます。





- テスト ロジックの追加
- ビジュアル テストの再生方法の指定
- 他のビジュアル テストおよびスクリプトの実行
- ビジュアル テストのデバッグ
- エラー処理の提供

テスト ステップ ペインをカスタマイズして、ビジュアル テスト内の必要なステップのみを表示できます。

結果 ウィンドウに表示される **テスト ステップ** ペインには、テスト結果に固有の追加情報が表示されます。**テスト ステップ** ペインが表示されない **要約** タブを除き、この情報は **基本表示** と **詳細表示** に表示されます。

テスト ステップ ペインの標準ビューには以下の列があります。

列	説明
 (番号) *	ステップが記録され、再生される順番を表します。デフォルトでは、ステップはこの順番で表示されます。

列	説明
	<p>ステップにデバッグのためのブレークポイントが追加されている場合は、この列の隣の列にブレークポイントが表示されます。</p> <p>デフォルトのビュー（ステップと画面）から画面のみのビューまたはステップのみのビューにビューを変更した場合、番号付けの方法では、記録の順序で間隔が示されるようにステップまたは画面は順番にスキップされます。</p>
 (ロジック) *	ステップにロジックが含まれている場合、ステップのロジックのタイプを表すアイコンを表示します。
 (ステップ タイプ) *	ステップで実行される操作のタイプを表すアイコンを表示します。
ステップ	ステップで実行される操作について説明します。
 (フラグ) *	<p>割り当てられたフラグ のアイコンを表示します。</p> <p>割り当てられたフラグ のアイコンは、関連付けられたステップが割り当て先のユーザーの テスト ステップ ペインおよびオプションで 開始画面 ペインに表示されるように、フラグが付けられていることを示します。さらに、ポインタをアイコンの上に移動すると、フラグの説明、更新日、および割り当て日を示すツールヒントを表示できます。</p>
 (ステップの説明) *	<p>ユーザー定義のステップの説明を表示します。この列はデフォルトのビューには表示されません。テスト ステップ ペインのタイトルバーから 操作 をクリックし、次に 表示 > ステップの説明 をクリックすると表示できます。</p> <p>ステップの説明を作成するには、ステップを選択します。プロパティ ペインで、ステップの説明 プロパティを更新します。ステップの説明 アイコンがこの列に表示され、ステップに説明があることが示されます。ポインタをアイコンの上に移動して説明を含むツールヒントを表示するか、ステップを選択してプロパティ ペインの説明を読みます。</p>

* この列はサイズ変更できません。

[結果] ウィンドウの [基本表示]

列	説明
結果	ビジュアル テストのステップまたはスクリプトのコード行について、実行ステータスを表示します。
結果の詳細	ビジュアル テストのステップまたはスクリプトのコード行について、実行ステータスの詳細を表示します。














[結果] ウィンドウの [詳細表示]

列	説明
行番号	スクリプトのコード行の行番号またはビジュアル テストのステップ番号を示します。[コマンドの詳細] と組み合わせて使用すると、特定のコード行またはステップ番号を見つけることができます。
コマンドの詳細	実際のコード行テキストを表示します。[行番号] と組み合わせて使用すると、テストスクリプトの特定のコード行またはビジュアル テストのステップ番号を見つけることができます。
日付/時刻	テスト ステップまたはコード行が実行されたときのタイムスタンプを表示します。
ミリ秒	テスト ステップまたはコード行の実行時間を表示します。

列	説明
名前	ビジュアル テストまたはスクリプトの名前を表示します。
ユーザー名	テスト実行者のログオン識別子を表示します。
マシン名	テストを実行するために使用したデバイスの固有の識別子を表示します。

テスト ステップ アイコン

アイコンは、ビジュアル テストのテスト ステップで一般的に実行される操作をグラフィカルに表したものです。グラフィックは、**テスト ステップ** ウィンドウの **テスト ステップ** 列に表示されます。以下の表は、一般的に使用される操作とそのアイコンを示しています。

ステップ タイプ	アイコン	説明
画面キャプチャ		再生中にこのステップに対して画面キャプチャが行われることを示します。
オブジェクト クラス	各種	各 UI オブジェクト クラスには固有のテスト ステップ アイコンがあります。たとえば、Window クラスのアイコンは  であり、TextField クラスは  です。
オブジェクトの待機		待機 プロパティが挿入されたことを示します。
判断ロジック		If ステートメント、Else If ステートメント、または End If ステートメントを実行します。
繰り返しテスト ロジック		一連のステップを指定した回数だけ実行します。
検証ロジック		ステップの 条件 プロパティに規定された条件を検証します。
エラー処理ロジック		エラー処理を実行します。
ビジュアル テスト		指定したビジュアル テストを再生します。
外部プログラム		指定した外部プログラムをステップの属性に従って起動します。
式		式を含みます。
結果コメント		結果コメントを含みます。
ラベル		ラベルを含みます。

テスト ステップの種類

ビジュアル テストの各ステップでは、テスト アプリケーションのコントロールに対する操作の実行、複合条件の評価、または単にビジュアル テストへのブックマークの挿入など、特定の操作が実行されます。**テスト ステップ** ペインの **ステップ** 列では、対応するステップが実行する操作の説明が記述されています。

一般的なテスト ステップは、テスト アプリケーションのコントロールに対して操作を実行しないテスト ステップです。ビジュアル テストで利用できる一般的なテスト ステップの種類を以下の表に示します。

ステップ テキストの内容	説明
<<開始>>および<<終了>>	ビジュアル テストの最初のステップと最後のステップです。ビジュアル テストで使用する受け渡し変数のプロパティは、<<開始>> ステップのプロパティに表示されます。
検証	ステップの 条件 プロパティに規定された条件を検証します。
If/Else/Else If/End If	ユーザー定義の条件を評価して条件の戻り値に応じて一連のテスト ステップを実行する判断ロジックを実行します。

ステップ テキストの内容	説明
	If テスト ステップがビジュアルテストに挿入されると、対応する End If テスト ステップも挿入されます。
ビジュアル テストの再生	ステップに関連付けられている再生設定プロパティを使用して、指定されたビジュアルテストを再生します。
.NET スクリプトの再生	ステップに関連付けられている再生設定プロパティを使用して、指定されたスクリプトを再生します。
プログラムの実行 式	ステップの コマンド プロパティの設定に基づいて、外部プログラムを起動します。 式を実行して評価し、その結果を変数に設定します。
結果コメント	ビジュアル テストの再生時に指定されたコメントを作成して、結果に送信します。コメントには、テキスト、式の値、アクティブ データ ファイル、または式の結果を使用でき、テストの 結果の詳細 タブの コマンドの詳細 列に表示されます。
ラベル'ラベル名'	ステップをラベルとして設定します。ステップのテキストは、ステップの ラベル プロパティに設定されるラベルの名前です。 ラベルを使用すると、テスト内の特定の領域やステップをブックマークし、テスト フローや再生を制御できます。
エラー処理	エラー処理を実行します。ステップの 実行する操作 プロパティによって、エラー処理が有効になっているときにエラーが発生した場合のテストの動作が決まります。エラー処理が有効になっているときにエラーが発生した場合、ステップの 資産 プロパティを使用して、別のテストを再生することができます。
メッセージ ボックス	再生中にテスト担当者に対してメッセージ ボックスを表示します。メッセージ ボックスの内容は、ステップに設定されているプロパティ値で決まります。
再生設定の設定	実行環境の再生設定を特定の値に変更します。この値は、テストステップの 再生設定 プロパティで設定されます。値は、そのビジュアル テストの再生に対してのみ、値がリセットされるまで有効です。
再生設定の取得	実行環境の再生設定を取得して、取得した設定を変数に設定できるようにします。
コントロールの "" プロパティの取得	空のステップを挿入し、そのステップを使用してテスト アプリケーションのコントロールからプロパティをキャプチャして、変数に設定します。 プロパティ ペインの 識別 ボタンを使用してコントロールをキャプチャし、コントロールの特定のプロパティを選択します。
ラベルへ移動	指定されたラベル ステップにジャンプして、再生します。
" が存在するまで待機	指定したコントロールが存在、または消滅するまで待機します。待機するコントロールは、リテラル、変数、式、またはアクティブ データとして表すことができます。タイムアウト プロパティで待機時間を指定します。
プロパティ " の値が " になるまで待機	指定したコントロールの指定したプロパティが指定した値になるまで待機します。待機するプロパティは、リテラル、変数、式、またはアクティブ データとして表すことができます。タイムアウト プロパティで待機時間を指定します。
プロパティ " の値が " と異なる値になるまで待機	指定したコントロールの指定したプロパティが指定した値と異なる値になるまで待機します。待機するプロパティは、リテラル、変数、式、またはアクティブ データとして表すことができます。タイムアウト プロパティで待機時間を指定します。
コントロール " の表示が安定するまで待機	指定したコントロールの表示が変更されず、位置も変わらなくなるまで待機します。Interval プロパティで表示が安定するまでに要する時間を指定します。タイムアウト プロパティで待機時間を指定します。

ステップ テキストの内容	説明
停止	再生を完全に停止します。通常、停止は「If」ステップなどの条件ロジックで使用されます。
遅延	ステップの遅延時間 プロパティで指定されている時間だけ再生を一時停止します。ステップの遅延タイプ プロパティは、時間の単位（秒またはミリ秒）を決定します。
タイマー番号の開始	タイマーを作成して開始します。 タイマーを使用して、ビジュアル テスト全体またはその一部の再生に要する時間を測ります。1 つのビジュアル テストで複数のタイマーを使用できます。
タイマー番号の停止	ステップの タイマー番号 プロパティで指定されたタイマーを停止します。 タイマーを使用して、ビジュアル テスト全体またはその一部の再生に要する時間を測ります。1 つのビジュアル テストで複数のタイマーを使用できます。
タイマーの再開	ステップの タイマー番号 プロパティで指定された、停止中のタイマーを再開します。
eCATT 引数の取得	SAP eCATT 引数コンテナから引数の値を取得します。この値は、eCATT スクリプトからビジュアル テストに渡されます。 このステップでは、SAP eCATT スクリプトから渡される、事前定義のインポート引数に変数を渡します。 Silk Test Workbench がインストールされているコンピュータに SAP eCATT をインストールすると、eCATT 変数値が取得可能になります。
eCATT 引数の設定	SAP eCATT スクリプトに渡す引数の値を設定します。 このステップでは、SAP eCATT スクリプトに渡すことのできる、事前定義のエクスポート引数に変数を渡します。 Silk Test Workbench がインストールされているコンピュータに SAP eCATT をインストールすると、eCATT 変数値が設定可能になります。
アクティブ データ ファイル " に値を直ちに保存します	ステップの実行時に、指定されたアクティブ データ資産に対して更新された関連データを保存します。
アクティブ データ ファイル " に値を保存しません	指定されたアクティブ データ ファイルから更新されたデータが、ステップの実行時点でファイルに保存されないようにします。このステップのあとで、「アクティブ データ ファイルの" に値を直ちに保存します」ステップが同じアクティブ データ ファイルに対して実行されると、このステップのあとで更新されたデータがファイルに保存されます。
グローバル変数の設定	グローバル変数名を指定し、値を初期化することによって、グローバル変数を定義します。
グローバル変数の内容の取得	グローバル変数の値を取得して、ビジュアル テストで使用するローカル変数に設定します。

[ビジュアル ナビゲータ] : [画面プレビュー]

画面プレビュー には、**テスト ステップ** ペインに表示された、コントロールと対話する各ステップについてキャプチャされたテスト アプリケーションのイメージが表示されます。キャプチャされたイメージとして、デスクトップ全体、アプリケーション ウィンドウ、またはアクティブなウィンドウのみを示すことができます。キャプチャされたイメージは、関連するステップが実行される前のアプリケーションの状態を示します。

画面プレビュー と **テスト ステップ** ペインは同期されています。コントロールと対話するステップを選択すると、**画面プレビュー** にテスト アプリケーション画面のイメージが表示され、選択したステップで自動処理されるコントロールが強調表示されます。強調表示されたステップがイメージにあるコントロールと

対話すると、そのコントロールの周りに四角形が表示されます。また、キャプチャされたイメージ内のコントロールをクリックすると、そのコントロールと対話する連続ステップが存在する場合は、その最初のステップが強調表示されます。

ビジュアルナビゲータ でビジュアルテストを操作する場合、テストアプリケーションにアクセスしなくても、**画面プレビュー** のイメージを使用してステップを作成できます。

ビジュアルナビゲータ の **テストステップ** ペインに何も表示されていない場合、**画面プレビュー** には記録の開始方法を示すイメージが透かし表示されます。

キャプチャされたテストアプリケーション画面の表示

テストアプリケーションのキャプチャイメージを **画面プレビュー** で表示するには、表示するキャプチャイメージを含むステップを **テストステップ** ペインで選択します。キャプチャイメージが表示されたら、イメージのサイズを変更したり、別のウィンドウにイメージを表示したりできます。

1. 別のサイズで表示するキャプチャイメージに対応するステップを選択します。
2. **画面プレビュー** から、**操作** をクリックして以下のコマンドのいずれかをクリックします。
 - **ズーム** : 既存の **画面プレビュー** ペインでキャプチャイメージを表示します。割合（キャプチャされたサイズの 10% から 300%）を選択するか、**ウィンドウに合わせる** をクリックしてイメージを **画面プレビュー** の現在のサイズに合わせます。
 - **新しいウィンドウで開く** : キャプチャされた画面を **画面プレビュー** ペインとは別の、サイズ変更可能なウィンドウに表示します。

[ビジュアルナビゲータ] : [プロパティ]

ビジュアルナビゲータ の **プロパティ** ペインには、ビジュアルテストの **テストステップ** ペインで選択したステップのプロパティか、ビジュアルテストまたはテストスクリプトの結果が表示されます。プロパティは名前と値のペアとして表示されます。プロパティの値は、リテラル値、変数、またはアクティブデータ値です。

ビジュアルテストの **プロパティ** ペインを使用して、選択したステップのプロパティに値を割り当てることができます。これにより、Silk Test Workbench によるテストアプリケーションとの対話、再生の管理、変数の使用、他の機能テストの呼び出しの方法について、正確な情報を調整できます。結果の **プロパティ** ペインは読み取り専用です。

プロパティ ペインには、以下のオプションを含むツールバーがあります。

カテゴリ	プロパティをプロパティカテゴリ別にグループ化します。
アルファベット	プロパティをアルファベット順に表示します。
説明	プロパティ ペインの下部にヘルプの説明パネルを表示します。

結果の **プロパティ** ペインには追加のツールバー ボタンがあります。

再生前のビジュアルテストのステッププロパティを表示/非表示	再生前にビジュアルテストに設定されていたステップのプロパティを表示します。
--------------------------------------	---------------------------------------

テストステップに関して表示されるプロパティは、ステップのタイプに基づきます。

ビジュアルテストの [プロパティ] ペインの表示

プロパティ ペインには、資産に関連付けられたプロパティと各プロパティの現在の設定がリストされます。**プロパティ** ペインを使用して、資産のプロパティ値を変更できます。複数の資産またはオブジェクトを選択すると、**プロパティ** ペインには、選択したすべての資産またはオブジェクトに共通のプロパティがリストされます。

プロパティの値を変更するには、プロパティ名に隣接するテキストボックスでプロパティの値を変更します。値が事前に定義されているプロパティの場合は、該当するリストボックスに表示される値の 1 つを選択します。

1. **テストステップ** ペインから、表示するプロパティが含まれるステップを選択し、**操作 > プロパティ** をクリックします（または F4 を押します）。
2. **アルファベット** ツールバー ボタンをクリックしてプロパティをアルファベット順で表示するか、**カテゴリ** ツールバー ボタンをクリックしてプロパティをカテゴリ タイプ別に表示します。

[ビジュアル ナビゲータ] : [ストーリーボード]



ストーリーボード ペインでは、テキストまたはイメージを一連のサムネイルで表示することによって、現在のビジュアル テストの全般構造がグラフィカルに表示されます。 **ストーリーボード** ペインは、ビジュアル テストの概要を視覚的に表示し、ビジュアル テストのフローの概要を示します。






サムネイルとは、テスト アプリケーションからキャプチャされた画面か、1 つのステップまたは一連のステップを説明するテキストを表す縮小サイズのイメージのことです。サムネイルは、アプリケーション画面をキャプチャするテスト ステップに関連付けられています。テキスト サムネイルは、テスト アプリケーションと直接対話を実行しないテスト ステップを表します。例として、<<開始>> と <<終了>> のステップや、他のビジュアル テストまたはスクリプトが呼び出されるステップがあります。

テスト ステップ ペインに表示されるステップは、論理グループを示す濃い青の実線で区切られています。このようにグループ化されたステップは、**ストーリーボード** ペインの各サムネイルに対応します。

サムネイルのサイズは、サイズ変更可能な **ビジュアル ナビゲータ** ウィンドウのサイズに対して相対的になっています。

ストーリーボード ペインには、以下の情報の要素が表示されます。

要素	説明
サムネイル イメージ	<p>ビジュアル テストのフローをグラフィカルに表します。イメージは、対応するテスト ステップを表す、テスト アプリケーションからキャプチャされたビットマップを表します。</p> <p>ビジュアル テストの開始と終了を示すために、各 [ストーリーボード] には、最初のステップとして <<開始>> サムネイル、最後のステップとして <<終了>> サムネイルが表示されます。</p> <p>ビットマップがないステップまたはステップのグループについては、サムネイルに説明テキストが表示されます。サムネイルにテスト アプリケーションのビットマップが含まれていない場合があります。対応する 1 つまたは複数のステップがテスト アプリケーションと直接対話しない（スクリプトを開くステップなど）場合や、テスト アプリケーションがビットマップでキャプチャされていない場合です。Silk Test Workbench でテスト アプリケーションから画面をキャプチャする方法をカスタマイズできます。詳細については、「記録の出力 : テスト オプション」を参照してください。</p> <p>[ストーリーボード] のサムネイルをクリックして、それをアクティブなストーリーボード項目にします。アクティブなストーリーボード項目に対応するステップが テスト ステップ ペインで強調表示されます。対応する画面が 画面プレビュー に表示されます。 プロパティ ペインのプロパティ情報も、アクティブなサムネイルを反映して変更されます。</p>
ステップ番号	<p>各サムネイルの上に対応するステップ番号が表示されます。たとえば、番号 5 から 8 がサムネイルの上に表示された場合、そのサムネイルは現在のビジュアル テストのステップ 5 から 8 を表します。</p>
アイコン	<p>以下のアイコンは、サムネイルに関連付けられたステップまたはステップのグループについての情報を提供します。</p> <ul style="list-style-type: none">  フラグ付きステップ  ユーザーに割り当てられたフラグ付きステップ

要素	説明
	 ロジックまたはエラー処理ステップ  検証ロジック ステップ さらに、 結果 ウィンドウの [ストーリーボード] には以下のアイコンが表示されます。  合格した検証ロジック ステップ  失敗した検証ロジック ステップ  注: [ストーリーボード] のサムネイルに同じタイプのステップが複数含まれている場合は、そのサムネイルに下にフラグが 1 つだけ表示されます。

ストーリーボードでのサムネイルの移動

ストーリーボードでサムネイルを移動するには、以下のキーボード操作を使用します。

- ストーリーボードで次のサムネイルに移動するには、右矢印 キーまたは 下矢印 キーを使用します。
- ストーリーボードで前のサムネイルに戻るには、左矢印 キーまたは 上矢印 キーを使用します。
- 最初のステップに戻るには、Page Up キーを使用します。
- 最後のステップに戻るには、Page Down キーを使用します。
- ビジュアル テストの先頭にある <<開始>> サムネイルに移動するには、Home キーまたは Ctrl +Home キーを使用します。
- ビジュアル テストの最後にある <<終了>> サムネイルに移動するには、End キーまたは Ctrl+End キーを使用します。

ビジュアル ナビゲータのレイアウトのカスタマイズ


ビジュアル ナビゲータ の 画面プレビュー ペイン、プロパティ ペイン、および ストーリーボード ペイン は、ドッキングしたり、非表示にしたり、サイズを変更したりできます。


1. **ビジュアル ナビゲータ** のそれぞれのペインおよび関連ウィンドウには、右上隅に押しピンが表示されます。押しピンをクリックするとペインが非表示になります。ペインは非表示になり、**ビジュアル ナビゲータ** ウィンドウの境界線にタブとしてドッキングされます。
2. ポインタをタブの上に置くと、ペインが一時的に表示されます。
3. タブをクリックすると、ペインが一時的に再ドッキングされます。
4. ペインを再表示して、押しピンをクリックすると、ペインが永続的に再ドッキングされます。
5. **ビジュアル ナビゲータ** 内でペインのサイズを調整するには、サイズ変更するペインの間にカーソルを置きます。マウスのカーソルが横方向または縦方向の分割カーソルに変わります。分割カーソルをクリックし、ドラッグして 2 つのペインのサイズを調整します。
6. **表示 > レイアウトの復元** をクリックすると、**ビジュアル ナビゲータ** のレイアウトはデフォルトビューに戻ります。

[結果] ウィンドウ

ビジュアル テストまたはスクリプトを再生したあと、再生結果を **結果** ウィンドウに表示することができます。

Summary **Passed** Failed Flags Details (16 steps)

Overall result for run 1 :  **Passed** No verifications were executed

Browser:  Internet Explorer 11 - Windows 7

Reason for abort: Not applicable

Latest run number: 1

Recent runs: 1

Visual test executed: TestResultOS

Visual test description:

Result description:

Scripts (number of times each ran): [Project A.TestResultOS.version 1 \(1\)](#)

Verifications passed: 0 / 0

Verifications failed: 0 / 0

Flags: 0

Start time: 13.03.2018 14:14:39

End time: 13.03.2018 14:14:46

Total time: 7 s

Steps run: 16

Playback setting: System Defaults Edit...

結果 ウィンドウには以下の機能があります。

ビジュアル ナビゲータ テスト再生の全体の様子がすばやくわかります。詳細については、「[結果] ウィンドウのビジュアル ナビゲータ」を参照してください。

[結果] ウィンドウの ツールバー 結果の内容の表示と種類をカスタマイズすることができます。詳細については、「[結果] ウィンドウのツールバー」を参照してください。

[結果] ウィンドウの タブ タブには、結果の内容を特定の種類にまとめられています。詳細については、「[結果] ウィンドウのタブ」を参照してください。

アセットブラウザ

アセットブラウザ を使用してテスト資産を管理します。**アセットブラウザ** では、データベース内の各種種類の資産を一元的に作成、管理、表示できます。任意の時点で Ctrl+Alt+A を押すか、**開始画面** の **タスク** ペインから **アセットブラウザ** をクリックして、**アセットブラウザ** を表示します。

ある種類の既存の資産項目を表示するには、**資産の種類** でその種類をクリックします。関連する資産のリストが右側に表示されます。資産の名前変更、削除、または複製を行うには、資産を右クリックして、適切なコマンドを選択します。資産プロパティを表示するには、資産を選択して **ファイル > 要約情報** を選択します。

新しい資産を作成するには、資産の種類を右クリックして **新規作成** を選択します。新しい資産が、現在アクティブなプロジェクトに追加されます。


テストの開発中に多数の資産が作成される可能性があります。必要な項目を見つけられるように、資産のビューをさまざまな方法で変更できます。

アセットブラウザ のコントロールには以下のものがあります。

- アクティブプロジェクト** アクティブ プロジェクトを設定します。これは、新しいテスト資産が追加されるデフォルトのプロジェクトです。アクティブ プロジェクトを変更すると、新しいアクティブ プロジェクトの資産が自動的に表示されます。
- 表示** 表示したい資産のプロジェクト名のチェック ボックスをオンまたはオフにして選択します。アクティブ プロジェクトの資産が常に表示されます。
- フィルタ** 資産の作成者、最終更新者、作成日時、最終更新日時などの設定した条件に基づいて、資産のフィルタ処理を行います。
- 資産の種類** リストから資産の種類を選択し、適用されたフィルタ処理に基づいて、選択されているプロジェクトで使用可能なテスト資産を表示します。詳細については、「[資産の種類](#)」を参照してください。
- 資産リスト** 適用されたフィルタ処理に基づいて、選択されているプロジェクトの選択された種類の資産を表示します。
- 項目を開く** 開く資産の名前を入力します。

アセットブラウザ ではこれらのコントロールを使用して資産を管理する他、以下の処理を行うことができます。




- 項目のリストでカテゴリ別に資産を並べ替えるには、基準とする列見出しをクリックします。並べ替えの順序を逆にするには、列見出しを再びクリックします。
- 表示する列を選択したり、列の順序を変更したりするには、列見出しを右クリックし、**列のカスタマイズ** を選択します。
- 項目に直接アクセスするには、**項目を開く** テキスト ボックスにその項目の名前を入力します。

 **注:** 多数の資産を管理するために、**アセットブラウザ** のインスタンスを複数開くことができます。開ける **アセットブラウザ** のインスタンス数は、最大 32 です。


[オブジェクトの識別] ダイアログ ボックス

オブジェクトの識別 ダイアログ ボックスを使用して、テスト対象アプリケーション (AUT) のオブジェクト (コンテナなどの表示されないオブジェクトを含む) のロケータを識別できます。テスト中にオブジェクト マップ エントリーを使用しているかどうかによって、ロケータは、オブジェクト マップ項目の名前か、選択したオブジェクトの XPath ロケータのいずれかになります。テスト中にオブジェクト マップ エントリーを使用している場合、オブジェクト マップの名前がオブジェクトの XPath ロケータの別名として使用されます。**オブジェクトの識別** ダイアログ ボックスを使用してオブジェクトのロケータを識別した場合、そのロケータはビジュアル テストと VB .NET スクリプトで使用できます。また、XPath ロケータまたはオブジェクト マップ エントリーが有効で、期待するオブジェクトに対応していることを検証するために、**オブジェクトの識別** ダイアログ ボックスを使用することもできます。

オブジェクトの識別 ダイアログ ボックスは、次のように複数の方法で起動できます。

- ツールバーで、**オブジェクトの識別** をクリックします。
 -  **注:** テスト アプリケーションを指定していない場合、**アプリケーションの選択** ダイアログ ボックスが開き、オブジェクトを識別するアプリケーションを指定できます。
- メニューで、**ツール > オブジェクトの識別** をクリックします。
 -  **注:** テスト アプリケーションを指定していない場合、**アプリケーションの選択** ダイアログ ボックスが開き、オブジェクトを識別するアプリケーションを指定できます。
- **開始画面** で、**入門ガイド** タブを選択して **オブジェクトの識別** をクリックします。
 -  **注:** **アプリケーションの選択** ダイアログ ボックスが開き、オブジェクトを識別するアプリケーションを指定できます。
- 既存のビジュアル テストで、変更するオブジェクトを参照するテスト ステップを選択します。**プロパティ** ペインを開き、**ロケータ** フィールドをクリックして、**コントロールの識別 > オブジェクトの識別ダイアログ** を選択します。

オブジェクトの識別 ダイアログ ボックスには、以下のコントロールがあります。

識別の開始	識別の開始 をクリックして、テスト アプリケーションでオブジェクト情報の記録を開始します。ロケータを記録する前に、テスト対象アプリケーションを特定の状態にするには、 識別の停止 をクリックします。
選択モード	識別するオブジェクトを選択する方法として、選択モードを定義します。 クリック モードの場合は、識別するオブジェクトをクリックします。 ホット キー モードの場合は、識別するオブジェクトにマウス カーソルを移動して、 キー入力 リスト ボックスで指定したキー入力の組み合わせを使用します。大抵は、メニューやコンボ ボックスなど、ユーザーがクリックしたときにのみ表示されるオブジェクトをキャプチャする場合に ホット キー モードを選択します。 ホット キー モードでは、オブジェクトを選択し、ホット キー入力の組み合わせを押すことによって、オブジェクトを閉じることなくロケータをキャプチャできます。
選択済みロケータ	キャプチャしたロケータが表示されます。 識別の開始 をクリックすると、 選択済みロケータ フィールドには、現在マウス カーソルが指している AUT 上のオブジェクトのロケータが表示されます。  注: ブラウザーでテストしている場合、 選択済みロケータ フィールドに実際にキャプチャするとロケータのみが表示される。
テスト	このボタンをクリックすると、テスト対象アプリケーションで選択したオブジェクトがハイライトされます。このボタンを使用すると、正しいオブジェクトが選択されることを検証します。
ロケータの詳細の非表示 / ロケータの詳細の表示	このリンクをクリックして、 ロケータの詳細 テーブルを表示または非表示にします。
[ロケータの詳細] テーブル	ロケータの詳細 テーブルには、テストしているアプリケーションのオブジェクトとその関連するクラス タイプ、ロケータ値、および属性が一覧表示され、アプリケーションの階層を確認できます。このテーブルを使用して、選択やオブジェクトに割り当てる属性を絞り込むことができます。たとえば、自動生成されたロケータは与えられたテストで使用するのに最適なロケータではない場合があります。 ロケータの詳細 テーブルを使用すると、ロケータで利用するのに、より関係のあるプロパティを見つけることができます。
オブジェクト マップの表示	オブジェクト マップ項目名を ロケータ 列に表示するには、このチェック ボックスをオンにします。オブジェクト マップ項目名は、コントロールまたはウィンドウに対して、コントロールやウィンドウのロケータではなく論理名 (エイリアス) を関連付けます。デフォルトでは、オブジェクト マップ項目名が表示されます。ロケータを使用するには、このチェック ボックスをオフにします。
完全ロケータの表示	完全ロケータ名を表示するには、このチェック ボックスをオンにします。オブジェクトに関連付けられた属性のみを表示するには、このチェック ボックスをオフにします。
プロパティの表示	ロケータの詳細 テーブルで選択したオブジェクトの属性および属性値を表示するには、このチェック ボックスをオンにします。このテーブルの属性を選択して、ロケータの識別に使用できます。[プロパティ] サブツリーを非表示にし、ロケータの詳細のみを表示するには、このチェック ボックスをオフにします。
貼り付け	このボタンをクリックして、 選択済みロケータ を適切なビジュアル テスト、スクリプト、またはオブジェクト マップにコピーします。

オブジェクトの識別 ダイアログ ボックスを使用してオブジェクトをキャプチャすると、以下のことを実行できます。

- **ロケータの詳細** ツリーを展開し、コンテキスト メニューを使用してアプリケーションの階層を表示します。

- コンテキストメニューを使用してサブツリーにあるオブジェクトのロケータを取得します。
- サブツリーのリンクをクリックして、追加のロケータの属性を表示します。
- サブツリーにあるオブジェクトを選択して、**選択済みロケータ** テキストボックスに表示されている既存のオブジェクトを置き換えます。
- ビジュアルテストの**プロパティ** ペインで、ロケータを既存のロケータにコピーします。
- ロケータを .NET スクリプトに直接コピーします。

スクリプトで、ロケータと共に使用するコードを指定する必要があります。ロケータだけがコピーされます。

- ロケータをオブジェクト マップのオブジェクト マップ項目にコピーします。

[コード] ウィンドウ

コード ウィンドウでは、スクリプトの記録、設計、および変更を行います。

コード ウィンドウにアクセスするには、既存のスクリプトを開くか、スクリプトを作成します。

Main ページ

コード ウィンドウを開くと、**Main** ページに基本テンプレートが入力されます。スクリプトの記録後、**Main** ページには記録したすべてのステップが表示されます。このページにコードを手動で入力することもできます。**Main** ページは、**ファイル** カテゴリの**プロパティ** ペインにも表示されます。

[プロパティ] ペイン

プロパティ ペインを使用して、アプリケーション構成、ファイル、パラメータ、およびスクリプト参照を追加、表示、または変更します。**プロパティ** ペインには以下のカテゴリがあります。

- **アプリケーション構成** : スクリプトのアプリケーション構成を表示するには、このカテゴリを展開します。
- **ファイル** : コード ウィンドウの **Main** ページと、追加したその他のファイルを表示するには、このカテゴリを展開します。**Main** ページには、記録したすべてのステップが表示されます。
- **パラメータ** : スクリプトの入力パラメータと出力パラメータを表示するには、このカテゴリを展開します。必要に応じて、入力パラメータと出力パラメータを追加、編集、および削除できます。
- **参照** : スクリプト ライブラリの参照を表示するには、このカテゴリを展開します。必要に応じて、参照を追加および削除できます。

スクリプトの [プロパティ] ペインの表示

プロパティ ペインを使用して、アプリケーション構成、ファイル、およびスクリプト参照を追加、表示、または変更します。コード ウィンドウの**プロパティ** ペインは、ドッキング、非表示、またはサイズ変更が可能です。

1. スクリプトを開きます。コード ウィンドウが開きます。
2. **プロパティ** ペインには、右上隅に押しピンが表示されます。押しピンをクリックするとペインが非表示になります。ペインは非表示になり、コード ウィンドウの境界線にタブとしてドッキングされます。
3. ポインタをタブの上に置くと、ペインが一時的に表示されます。
4. タブをクリックすると、ペインが一時的に再ドッキングされます。
5. ペインを再表示して、押しピンをクリックすると、ペインが永続的に再ドッキングされます。
6. ペインのサイズを調整するには、サイズ変更するペインの間にカーソルを置きます。マウスのカーソルが横方向または縦方向の分割カーソルに変わります。分割カーソルをクリックし、ドラッグして2つのペインのサイズを調整します。

出カウィンドウ

出カ ウィンドウには、スクリプトの再生中に見つかったエラーに関する情報、スクリプトで実行された検索の結果、スクリプトによって書き込まれたテキストなどが表示されます。

出カ ウィンドウには、参照されているオブジェクト マップ エントリが実行中の資産の範囲内でない場合、再生に関する警告メッセージも表示されます。



注: Pushbutton.Select などの NTF メソッド (呼び出し) の結果のみがログに記録されます。したがって、with ステートメントなどの VB.NET コマンドを実行した場合、VB.NET 呼び出しも _desktop.Window() 呼び出しもログに記録されません。

既知の問題

このセクションでは、Silk Test Workbench の既知の問題とその解決策を示します。

全般的な問題

オブジェクト マップを開くのに時間がかかる

.NET 4 を使用している場合、大規模なオブジェクト マップ資産があると読み込みに時間がかかります。.NET 4.5 をインストールすると、この問題を解決できます。

リモート デスクトップまたはリモート デスクトップ接続 (RDC) が最小化されると、Silk Test が機能しない

リモート デスクトップ プロトコル (RDP) 経由でデスクトップに接続している場合、マウスとキーボードを使ってデスクトップに接続することにより、デスクトップの所有権を獲得します。デスクトップの所有権を解放せずにデスクトップを最小化すると、マウス クリックまたはキーストロークの再生がすべて未定義になります。

回避策として、VNC ベースのリモート操作ツールを使用することができます。この場合、クライアントが最小化された場合でも再生を継続できます。

Check Point ファイアウォールがインストールされている場合に Open Agent が起動しない

システムに Check Point ファイアウォールまたは Check Point ZoneAlarm ファイアウォールをインストールしている場合は、ファイアウォールが Agent と infoservice 間の通信を中断するために Open Agent を起動できません。

Open Agent を起動するには、システムから Check Point ファイアウォールをアンインストールする必要があります。

domDoubleClick メソッドの modifiers パラメータが無視される

domDoubleClick メソッドのオーバーロードで修飾キーを指定できません。パラメータが指定されている場合でも、修飾キーはダブルクリックされません。修飾キーを指定できる domDoubleClick メソッドのオーバーロードは、非推奨です。修飾キーを指定するには、doubleClick メソッド (modifiers パラメータを取るオーバーロード メソッドをサポートするクライアントを使用している場合)、または PressKeys および ReleaseKeys メソッドを使用します。

Silk Test が Metro スタイル アプリのテストをサポートしない

Silk Test は、Microsoft Windows 8、Microsoft Windows 8.1、Microsoft Windows 10 上の Metro スタイル アプリのテストをサポートしません。Metro スタイル アプリは、Windows 8 スタイル、Modern UI スタイル、Windows ストア スタイル、ユニバーサル Windows プラットフォーム (UWP) アプリとも呼ばれます。

Microsoft Windows 8 の組み込みスペル チェックがテストの再生に干渉する

Microsoft Windows 8 の組み込みスペル チェックは、Internet Explorer 10 などのアプリケーションで有効にできます。

記録中に単語のスペルを間違え、この単語の入力を再生すると、スペルチェッカーはこの単語をマークするか、間違われやすい単語の場合は自動的に修正します。これは、実際のユーザーに対して行われる動作と同じです。スペル チェック機能が含まれていないオペレーティング システムでテストを作成した場合、

Microsoft Windows 8 でテストを再生すると、予期せぬ結果が生じることがあります。スペル チェックを無効にするには、次の手順を実行します。

1. **Windows キー + C** を押します。
2. チャーム バーで **設定** をクリックします。
3. **PC 設定の変更** を選択します。
4. **全般** を選択すると、スペル チェック セクションが表示されます (使用中の言語によってスペル チェック機能の有無は異なります)。



注: これはシステム全体の設定で、Internet Explorer 固有の設定ではありません。

5. **スペル ミスの語句を自動修正する (Autocorrect misspelled words)** をオフに設定します。
6. **スペル ミスの語句を強調表示する (Highlight misspelled words)** をオフに設定します。

.NET アプリケーションを DevPartner Studio (DPS) から起動すると、Silk Test で認識されないことがある

この問題を解決するには、以下のステップを実行します。

1. Silk Test のインストール フォルダに移動します (デフォルトでは C:\Program Files\Silk\SilkTest)。
2. Windows Forms アプリケーションの場合は、ng\agent\plugins\com.borland.fastxd.techdomain.windowsforms.agent_<バージョン番号> へ移動します。
3. Windows Presentation Foundation (WPF) の場合は、ng\agent\plugins\com.microsoft.silktest.techdomain.wpf.agent_<バージョン番号> へ移動します。
4. メモ帳で、plugin.xml ファイルを開いて、<loadparameters> セクションに以下の行を追加します。

```
<param name="frameworkAssembly">mscoree.dll</param>
```
5. コンピュータからログアウトして、再びログインします。DevPartner Studio によって起動されたアプリケーションに対して、Silk Test が期待どおりの動作をします。

イメージ領域に対するクリックの記録時に矩形領域によってハイライトされる位置がずれる

エリア マップのような複雑なイメージの一部に対するクリックを記録する場合、イメージの適切な領域が緑色の矩形領域によってハイライトされません。ただし、再生時にクリックは正しく実行されます。

Silk Test のインストール中に Windows Defender が有効になっていると Open Agent が起動しない

Silk Test のインストール中にシステムの Windows Defender が有効になっていると、インストールが完了した後に Open Agent を起動できなくなる場合があります。Windows Defender は、ホットフィックス セットアップに必要な操作を妨げる場合があります。回避策として、Silk Test のインストール中は Windows Defender を無効にしてください。

IME エディタが特定の場所から開かれたときにポップアップする

IME エディタを開くと、テキスト フィールド内には開くのではなく、現在の画面の左上隅にポップアップして開きます。

この動作は、次の場所から IME エディタを開いたときに発生します。

- **Silk Recorder**
- **キーワード駆動テスト エディター**
- **キーワード ウィンドウ**

検索ボックスに文字を入力すると Silk Test ヘルプがフリーズする

Microsoft Windows 8.1、Microsoft Windows RT 8.1、または Microsoft Windows Server 2012 R2 上のコンパイルされた HTML ヘルプ (.CHM) ファイルで、**キーワード** タブの **検索** ボックスに文字を入力すると、Silk Test ヘルプがフリーズする場合があります。この問題を解決するには、Windows Update をイ

インストールする必要があります。このアップデートは <https://support.microsoft.com/kb/3080042> から利用できます。

Num Lock キーがオンの状態で Shift + Insert を使用できない

Left Shift + Insert や **Right Shift + Insert** を使用して TypeKeys メソッドでクリップボードの内容を貼り付ける操作は、Num Lock キーがオンの場合に機能しません。

TypeKeys メソッドを呼び出す前にテストスクリプトで Num Lock キーをオフにすることで、この問題を回避できます。

「Windows DLL failed to load」というエラーでインストーラが異常終了する理由

Java 8 の Java 8 update 212 以降を使用すると、Silk Test を Windows Server 2019 上にインストールするときに次のエラーで失敗します : 「Windows DLL failed to load」。

update 212 より前のバージョンの Java 8 をダウンロード、インストールして、次のようなコマンドでインストーラを実行することによって、この問題を回避できます。

```
SilkTest<Version>.exe LAX_VM "[path to java.exe]"
```

例 :

```
SilkTest200_64bit.exe LAX_VM "c:¥java¥bin¥java.exe"
```

コマンドラインから Silk Test をインストールする方法についての詳細は、『[Silk Test インストールガイド](#)』を参照してください。

モバイル Web アプリケーション

Apple Safari のフレームを Silk Test Workbench がサポートしない

Silk Test Workbench は、iOS 上の Apple Safari を使用した HTML フレームおよび iframe のテストをサポートしません。

Chrome for Android 43 以降 : 拡大縮小とスクロールを同時に Silk Test Workbench がサポートしない

Chrome for Android 43 以降でモバイル Web アプリケーションを記録しているとき、モバイル Web アプリケーションが拡大されて左上隅が画面から見えなくなると、期待通り機能しない場合があります。モバイル Web アプリケーションのコントロールに対して緑の矩形領域が記録時に正しく表示されない場合、モバイル Web アプリケーションを完全に縮小し、モバイル Web アプリケーションの左上隅にスクロールして、記録 ウィンドウを更新してください。

Web アプリケーション

100% 以外の拡大レベルを使用して記録すると期待通り機能しない可能性がある

100% 以外の拡大レベルを使用して Web アプリケーションを記録すると、期待通り機能しない可能性があります。Web アプリケーションに対する操作を記録する前に、ブラウザの拡大レベルを 100% に設定してください。

Google Chrome

Windows で記録中のロケーターが Google Chrome で失敗する

Google Chrome で Web アプリケーションをテストするときに、アプリケーションを実行している Google Chrome インスタンスのアプリケーション構成中に複数のウィンドウが開いていると、Windows

で記録中のロケーターが失敗します。アプリケーション構成中に他の Google Chrome ウィンドウを閉じると、エラーは発生しなくなります。

Google Chrome のバックグラウンド アプリケーションでオートメーション サポートを読み込むことができない

Google Chrome を使用して Web アプリケーションをテストしている場合に、**Google Chrome を閉じた際にバックグラウンド アプリケーションの処理を続行する** チェックボックスがチェックされていると、Silk Test は Google Chrome を再起動してオートメーション サポートを読み込むことができません。

Windows Aero が無効なときにモーダル ダイアログのロケーターを Silk Test が記録できない

Windows Aero が無効化されている場合、モーダル ダイアログが認識されないため、このようなダイアログのロケーターを選択できません。回避策として、モーダル ダイアログが表示されているときには、**Locator Spy** または **オブジェクトの識別** ダイアログ ボックスを使用して、ロケーターを手動で作成および検証してください。

Silk Test が埋め込み PDF を表示しない

Google Chrome 42 以降を使用すると、Google Chrome は、埋め込み PDF を表示するために使用する NPAPI プラグインをデフォルトでブロックします。このため、Silk Test は埋め込み PDF を Google Chrome 42 以降に表示する代わりに、埋め込み PDF をダウンロードします。

- Google Chrome 44 以前を使用している場合、アドレス バーに以下を入力することにより、Google Chrome での NSAPI プラグインのブロックを解除できます。
`chrome://flags/#enable-npapi`
- Google Chrome 45 以降を使用している場合、NPAPI プラグインは Google Chrome から完全に削除されており、再有効化するオプションも無いため、すべての PDF はダウンロードされます。

テストの実行時に Google Chrome (49 以前) との接続がタイムアウトする

低速なマシンで Google Chrome (49 以前) に対するテストを実行すると、接続タイムアウトが発生し、テストが失敗する場合があります。以下のエラー メッセージが表示されます。

'*' の実行中にエラーが発生しました。ブラウザ オートメーションとの通信がタイムアウトしました。

このような接続タイムアウトを回避するために、十分な処理速度を持つテスト マシンを使用してください。たとえば、低速の仮想マシン (VM) 上でテストしている場合には、VM に CPU コアを追加することで処理速度を向上させることができます。

Google Chrome 66 以降を使用している場合にレジストリを使って UserDataDir を設定すると Google Chrome サポートが機能しなくなる

レジストリのポリシーとして HKEY_LOCAL_MACHINE¥Software¥Policies¥Google¥Chrome ¥UserDataDir キーや HKEY_CURRENT_USER¥Software¥Policies¥Google¥Chrome¥UserDataDir キーを使ってユーザー データ ディレクトリを設定すると、Google Chrome 66 以降上で Web アプリケーションをテストする際に、基本状態の実行が次のエラー メッセージで失敗する可能性があります:「アプリケーション 'GoogleChrome' の開始に失敗しました。 unknown error: DevToolsActivePort file doesn't exist」。これは ChromeDriver の既知の問題です (<https://bugs.chromium.org/p/chromedriver/issues/detail?id=2513>)。

この問題を回避するには、以下のいずれかのステップを実行します。

- レジストリ キーを削除します。



注: この問題が発生した場合に、Google Chrome を使って並列テストを実行するには、この方法の適用が必須です。

- Google Chrome のユーザー データ ディレクトリをレジストリ キーと同じディレクトリに設定します。

1. **アプリケーション構成の編集** ダイアログ ボックスを開きます。
2. ブラウザーの種類から **Google Chrome** を選択します。
3. ユーザー データ ディレクトリを **接続文字列** フィールドに設定します：
goog:chromeOptions={"args":["--user-data-dir=<ユーザー データ ディレクトリ>"]}たとえば、レジストリの値が C:/temp/chromeUserData であれば、goog:chromeOptions={"args":["--user-data-dir=C:/temp/chromeUserData"]} と入力します。



注: Google Chrome を使った並列テストでは、この回避策は機能しません。

Internet Explorer

Google ツールバーを使用すると、Web アプリケーションの記録に支障をきたす

Internet Explorer 8 で Google ツールバーを使用すると、Web アプリケーションのロケーターの記録に支障をきたします。

Google ツールバーをオフにしてから、Web アプリケーションを記録してください。

Microsoft Silverlight アプリケーション

一部の Microsoft Silverlight アプリケーションで、Silk Test との通信の際に Internet Explorer がハングします。32 ビットプラットフォームでは、問題の防止に役立つ MS KB 2564958 (Active Accessibility の更新プログラム) を参照してください。

Silk Test 13.5 より前のバージョンの Silk Test を使用して記録したロケーターが Internet Explorer で動作しない

Silk Test 13.5 で、Internet Explorer の textContents 属性における空白文字の標準化を改良しました。この変更は、Silk Test のクロスブラウザ機能を改善するための措置で、textContents 属性を利用しているロケーターに影響を与える可能性があります。この属性は、Silk Test 13.5 以前のリリースを使用して記録されたスクリプトで使用されています。

UAC が Microsoft Windows 8 以降および Internet Explorer 11 で有効化されているとき、Open Agent に対して高い昇格を有効化できない

UAC が有効化され、Internet Explorer と Open Agent の両方を高い昇格で実行するとき、Microsoft Windows 8 以降で Internet Explorer 11 上の Web アプリケーションをテストできません。

入力方式エディター (IME) における既知の問題

- Silk Test は、Internet Explorer 11 での日本語入力において、半角スペースを記録しません。
- Silk Test は、Internet Explorer 11 の互換モードでの IME 入力を記録しません。
- 日本語 IME モードで、**スペース** を押すと、Silk Test は現在の IME の候補を記録することがあります。この問題を避けるには、**変換** を使用してください。

Java 1.7 update 71 よりも新しいバージョンを使ったアプレットのテスト時に Internet Explorer が動作を停止する場合がある

Java 1.7 update 71 (7u71) よりも新しいバージョンを使ってアプレットをテストすると、Internet Explorer が動作を停止する場合があります。

テストの実行中に Internet Explorer が応答しなくなる

Internet Explorer 10 以降でテストを実行していると、Internet Explorer のスレッドが中断され、デッドロックが発生したことが原因で、応答しなくなる場合があります。この問題は、Internet Explorer の新しいセキュリティ機能が原因です。

この問題を解決するには、セキュリティ機能を無効化します。

1. レジストリ エディタ が開きます。
2. `OverrideMemoryProtectionSetting` エントリが `HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Main` キーに存在しない場合は、エントリを作成します。
3. レジストリ キー `HKEY_CURRENT_USER\SOFTWARE\Microsoft\Internet Explorer\Main::OverrideMemoryProtectionSetting` の値を 2 に設定します。

JavaScript 警告処理 API メソッドが埋め込み Internet Explorer で機能しない

`BrowserWindow` クラスの以下の JavaScript 警告処理メソッドが、埋め込み Internet Explorer のテスト時に機能しません。

- `AcceptAlert` メソッド
- `DismissAlert` メソッド
- `GetAlertText` メソッド
- `IsAlertPresent` メソッド

Microsoft Edge

Microsoft Edge から開いたウィンドウがサポートされない

タブではなく、Microsoft Edge の新しいウィンドウとして開いたウィンドウはサポートされません。Silk Test Workbench は、このようなウィンドウを正しく閉じることができず、このようなウィンドウを閉じると、エージェントの状態が不正になります。

iframe または frame で開いた JavaScript 警告を閉じることができない

JavaScript 警告 `alert()`、`prompt()`、`confirm()` が、`iframe` または `frame` で開かれた場合、Microsoft Edge は閉じることができません。

ネイティブ ブラウザーのコンテキスト メニューを開けない

Microsoft Edge を右クリックして、ネイティブ ブラウザーのコンテキスト メニューを開く操作がテストに含まれていると、そのテストはハングします。Microsoft Edge によって開かれた HTML メニューでは、この問題は発生しません。

UAC を無効にした状態で Microsoft Edge に対するテストを実行できない

UAC を無効にした状態で Microsoft Edge に対してテストを実行すると、次のエラー メッセージが表示されます。

```
Failed to start application 'Edge'. Unable to parse remote response: Unknown error
```

この問題を解決するには、UAC を有効にしてください。

「Windows Feature Microsoft WebDriver がインストールされていないため、Edge を開始できません」エラーが表示される理由

Microsoft Windows 10 の Windows 10 October 2018 Update 以降のバージョンを使用している場合、オプション機能「Windows Feature Microsoft WebDriver」がシステムにインストールされていないと、Microsoft Edge 上でのテストの実行時に次のエラーが発生する場合があります。

Windows Feature Microsoft WebDriver がインストールされていないため、Edge を開始できません。ヘルプのトピック「既知の問題」>「Microsoft Edge」を参照してください。

この問題を解決するには、システムで Windows Update が正しく機能していることを確認してください。内部の Windows Update サーバーを使用している場合は、次の手順を実行する必要があります。

1. IT 管理者に依頼して、Windows Feature Microsoft WebDriver を Windows Update サーバーに追加してもらいます。

2. Windows Feature Microsoft WebDriver をインストールします。オプション機能のインストールについては、「[Microsoft WebDriver](#)」を参照してください。

Mozilla Firefox

Adobe FlashPlayer を使用したアプリケーションの呼び出しが Mozilla Firefox を使用した場合に正しく同期されない

最近のバージョンの Adobe FlashPlayer を Mozilla Firefox で使用している場合、呼び出しが正しく同期されないことがあります。次の問題が発生する可能性があります。

- Mozilla Firefox がスクリプトの実行がハングしたものと誤って認識し、スクリプトが正しく実行されているにもかかわらず、スクリプトの実行を続けるかどうかを確認するダイアログ ボックスが表示される場合があります。
- SetFocus が正しく機能しないため、文字の入力が機能しない場合があります。
- UI には新しい値が表示されているにもかかわらず、Adobe オートメーションが古い値を返す場合があります。

Adobe FlashPlayer を使用するアプリケーションでこのような問題に遭遇した場合は、Adobe FlashPlayer の ProtectedMode をオフにしてください。詳細については、<http://forums.adobe.com/thread/1018071> を参照し、Last Resort に記述された情報をお読みください。

SAP アプリケーション

SAPTree クラスの HierarchyHeaderWidth および ColumnOrder プロパティが書き込み専用になっている

自動化に関するドキュメントに記載されている場合を除いて、SAPTree の HierarchyHeaderWidth および ColumnOrder プロパティは書き込み専用で、読み込むことはできません。

これらのプロパティを使用する場合、スクリプトで、読み込みではなく、書き込みが使用されていることを確認します。

SAPTree クラスの GetColumnIndexFromName() が「特定できないエラー」により失敗する

SapTree クラスの GetColumnIndexFromName() は「特定できないエラー」により失敗することがあります。これは SAP オートメーションの既知の問題です。

SAP Web サイトでこの問題が解決されているかどうか確認してください。

コンテキスト メニュー項目の SAPTree クラスの Select() メソッドの呼び出しに失敗する

コンテキスト メニュー項目の SAPTree クラスの Select() メソッドの呼び出しに失敗することがあります。

代わりに親コントロールの SelectContextMenuItem を呼び出します。これは SAP オートメーションの既知の問題です。

水平スクロールバーの Position プロパティが常に 1 を返す

水平スクロールバーの Position プロパティは常に 1 を返します。これは SAP オートメーションの既知の問題です。

SAP Web サイトでこの問題が解決されているかどうか確認してください。

SAPNetPlan クラスがサポートされていない

この問題は今後のリリースで解決される予定です。

SAP スクリプトを実行すると、再生エラーが発生する

ある状況下で、SAP テストを記録して再生すると、次のエラーが発生する場合があります：この操作を完了するのに必要なデータは、まだ利用できません。このエラーは、Silk Test Workbench が記録した操作の実行が速すぎたことを意味しています。

この問題を解決するには、テスト スクリプトにスリープを追加したり、関数の呼び出しやプロパティの設定の後に Silk Test Workbench が待機する時間を増やすために、再生後の遅延を増やしてください。このオプションについての詳細は、「エージェント オプション」を参照してください。また、xBrowser テクノロジドメインではなく、SAP オートメーションを使用して問題が発生する操作を再生するように、スクリプトを変更します。たとえば、その操作を DomLink.Select から SapHTMLViewer.SapEvent に変更します。

SAPGUI クライアント 7.30 を使用するとメソッド getCurrentRow が誤った値を返す

SAPGUI クライアント 7.30 を使用しているときに、メソッド getCurrentRow を呼び出すと、メソッドは行番号ではなく誤った値 -1 を返す場合があります。

メソッド resizeWorkingPane が SAPGUI クライアント 7.30 で正しく機能しない

SAPGUI クライアント 7.30 を使用しているときに、メソッド resizeWorkingPaneEx を呼び出すと、メソッドは workingPane のサイズを変更せずに、getSapWindow().getWidth() の呼び出し結果は、ウィンドウの幅と異なる値を返します。

Silk Test Workbench を eCATT から開始すると仮想マシン上で記録を開始できない

SAPGUI から eCATT スクリプトを作成したとき、記録 ボタンを押して仮想マシンの操作を記録しようとしても、VB .NET スクリプト、ビジュアル テスト、キーワード駆動テストが Silk Test Workbench エディタで開いていると、動作しません。物理マシン上での記録操作は動作します。

eCATT から Silk Test Workbench データベースに資産インポートするとき、エクスポートした資産の XML よりも新しいバージョンの Silk Test Workbench で作成された eCATT BLOB を使用してもエラーが表示されない

eCATT から Silk Test Workbench データベースに資産をインポートするときに、Silk Test Workbench のバージョンが eCATT BLOB の資産を含む XML のバージョンよりも古い場合、インポート処理はエラーを表示せずに失敗します。

インポートが失敗すると、次に BLOB を編集しようとしたときに、「eCATT に戻る前に現在の BLOB を保存しますか?」というメッセージが表示されます。この場合、BLOB は空のため保存してはいけません。保存せずに、**いいえ** をクリックして eCATT に戻ってください。



ヒント: eCATT を使用して Silk Test Workbench の複数のインスタンスを使用している場合、Micro Focus では、すべてのインスタンスを同じ Silk Test Workbench のバージョンに合わせることを推奨しています。

Silk Test Workbench

大規模サイトでのビジュアル テストの再生

xBrowser の使用中に大規模サイトでビジュアル テストを実行すると、パフォーマンス問題が発生することがあります。この問題を回避するには、**再生 > 結果 > ビジュアル テスト > コントロールのキャプチャ オプション**を **いいえ** に設定します。

スクリプトの再生時にバックグラウンドにメッセージ ボックスが表示される (31314)

メッセージ ボックスのステートメント (MsgBox ("Hello") など) を .NET スクリプトに含めた場合、スクリプトの再生時にバックグラウンドにメッセージ ボックスが表示されます。

メッセージ ボックスをフォアグラウンドに表示するには、MsgBoxStyle.MsgBoxSetForeground ステートメントに MsgBox を含めます。例：

```
MsgBox ("Hello", MsgBoxStyle.MsgBoxSetForeground)
```

インストーラが SQL Server Native Client のインストールと COM サーバーとしての Silk Test Workbench の登録に失敗する

Silk Test のインストール中またはインストール前に、オペレーティング システムが自動更新を実行するか、またはユーザーが Windows アップデートを行った場合、インストーラは SQL Server Native Client のインストールと COM サーバーとしての Silk Test Workbench の登録に失敗します。

これらの問題は共に、Silk Central Test Manager (SCTM) の統合に影響を与えます。それは、SCTM では SQL Server Express によってインストールされたネイティブ クライアント ドライバを使用して Silk Test Workbench DSN を作成するためです。さらに SCTM は、自動化に COM インターフェイスを使用するには、COM サーバーとして登録された Silk Test Workbench を必要とするためです。

以下の解決策のいずれかを選択します。

1. Windows アップデートの実行中に Silk Test をインストールしない。
2. 使用しているマシンに Windows アップデートが自動的にインストールされないようにする。
3. 保留中の Windows アップデートを Silk Test より先にインストールしてから、システムを再起動し Silk Test をインストールする。

複数の修飾キーを含めるようホット キーの組み合わせを変更すると、追加のキーが記録される

検証の挿入用または記録の開始/停止用に複数の修飾キーを含むホットキーを構成している場合は、修飾キーを押す操作が記録されることがあります。たとえば、Alt+Ctrl+F9 というホット キーの組み合わせを指定した場合、Alt は、ホット キーの組み合わせの一部として無視されるのではなく、キーを押す操作として記録されることがあります。これは、キーから手を離すという対応する操作がないため、再生時に問題になります。

修飾キーに関連するこの操作を手動で削除するか、ホット キーの修飾キーを 1 つだけ使用します。

データ ソース名 (DNS) が 64 ビット システムで動作しない

SQL Server または Oracle の 64 ビット DSN を Silk Test Workbench に使用することはできません。64 ビット マシンで DSN を作成するには、**スタート > Silk > Silk Test > 管理 > データ ソース (ODBC)** をクリックして、32 ビット DSN を作成します。C:\¥WINDOWS¥SysWOW64¥odbcad32.exe にある WOW64 ツールを使用することもできます。

Silk Test Workbench を起動するのにネットワーク アダプタが必要になる

Silk Test Workbench を開始するときに、Silk Test Workbench と Open Agent の間で通信できるようにするため、Silk Test Workbench をインストールしたマシンでネットワーク アダプタが使用可能である必要があります。ネットワーク アダプタが使用できない場合、エラー「Open Agent との接続に失敗しました」が発生します。他のクライアント (Silk Test Classic や Silk4) を起動する場合には、ネットワーク接続は必要ありません。

Silk Test Workbench を起動する前に、ネットワーク アダプタが使用可能であることを確認してください。

SQL Server Express を Windows Vista SP1 マシンにインストールできない

Windows Vista SP1 マシンでは、SQL Server 2008 Express をインストールしようとする場合、.NET 3.5 SP1 の完全バージョンがインストールされていることを確認してください。.NET 3.5 SP1 の完全バージョンをインストールするには、Microsoft の Web サイトからダウンロードします。

[画面プレビューから識別] により、移行されたデータベースに不正なオブジェクト マップ項目が作成される

Silk Test Workbench 2010 から Silk Test Workbench 2010 R2 に移行したあと、**画面プレビューから識別** を使用すると不正なオブジェクト マップ項目が作成されます。

画面プレビューから識別 を使用する前に、**画面の更新** コマンドを使用して、画面を再キャプチャします。

子スクリプトが親スクリプトから呼び出されると、子スクリプトはスコープ外のオブジェクト マップにアクセスできる

親スクリプトと子スクリプトが異なるプロジェクトに含まれており、異なるオブジェクト マップを使用している場合、オブジェクト マップが子スクリプトのスコープ外であっても、子スクリプトは正常に実行されます。

スクリプトが正常に実行されるため、解決策は必要ありません。ただし、この動作は今後のリリースでは正常に実行されなくなります。

複数ユーザーが SQL 2008 データベース上で同じ資産を同時に編集しようとする、資産のロック機構に関する問題が発生する可能性がある

問題かどうかをテストするには、ビジュアルテストなどの資産を開いて、Silk Test Workbench で編集できるようにします。資産が開いている間に、Silk Test Workbench の別のインスタンスから同じ資産を開きます。問題が発生した場合は、Silk Test Workbench の 2 番目のインスタンスで同じ資産を同時に編集できるか、または 2 番目のインスタンスが不完全なデータに関するメッセージを受け取ります。このときユーザー名とマシン名は共に空白になります。

SQL Server で、データベースの管理者権限を、このデータベースを使用する SQL データベース ユーザーに割り当てます。

SQL データベースで資産のエクスポート中に予期しないエラーが発生する

SQL データベースで資産のエクスポート中に予期しないエラー「文字列から uniqueidentifier に変換中、変換に失敗しました」が発生します。

エクスポートしたデータベースをインポートし、何らかのエラー メッセージが表示された場合 OK を押します。インポートは正常に終了し、資産を使用できます。

Access データベースの使用中に予期しないエラーが発生する

Silk Test **データベース メンテナンス** ツールを使用してデータベースをコンパクト化します。(Microsoft Windows 7) **スタート > すべてのプログラム > Silk > Silk Test > 管理 > データベース メンテナンス**、または (Microsoft Windows 10) **スタート > Silk > データベース メンテナンス** をクリックします。次に、データベースを開き、**ツール > データベースのコンパクト化** をクリックします。

構成中に UAC が有効な状態で Silk Test Workbench が SQL Server へ接続できない

構成中に管理者権限を持たずに Silk Test Workbench を使用している場合、UAC が有効な状態で SQL Server へ接続できません。

構成中に SQL Server へ接続するには、**管理者として実行** オプションを使用して Silk Test Workbench を起動する必要があります。

リモートの SQL データベースとの接続が失われたときに Silk Test Workbench が異常終了する場合があります

リモートの SQL データベースを使用している場合、データベースが Silk Test Workbench から切断されると (サーバーを再起動した場合など)、Silk Test Workbench がデータベースとの通信を行おうとする際に異常終了する場合があります。

Silk Test Workbench を再起動して、データベースと再接続してください。

コントロールのキャプチャ オプションが設定されているとビジュアル テストの再生速度が著しく低下する 場合がある

再生のパフォーマンスを向上させるには、コントロールのキャプチャ オプションを **いいえ** に設定します。

Microsoft Windows 7 クラシック テーマ : ビジュアル テストの画面プレビューに 記録中 ウィンドウが 含まれる場合がある

クラシック テーマを使用した Microsoft Windows 7 マシンで記録したビジュアル テストの画面プレビューに **記録中** ウィンドウが含まれる場合があります。Microsoft Windows 7 で **記録中** ウィンドウを含まずにビジュアル テストを記録するには、クラシック テーマを使用しないでください。

資産名の変更時に参照を更新しない場合に発生する問題

ほかの資産から参照されている資産の名前を変更する時に、参照も更新するようにしてください。同時に参照を更新しなかった場合、資産の名前を元に戻して、再度名前を変更しなおしても、Silk Test Workbench は参照を更新を確認するダイアログを表示しないため、参照を更新できません。

名前を変更する資産が .NET スクリプトで、.NET スクリプトを参照している資産がビジュアル テストの場合は、ビジュアル テストを保存すると、この問題を回避できます。保存することにより、Silk Test Workbench は .NET スクリプトを見つけ出し、資産の参照が正しくなるよう更新されます。

Oracle 19 のサポート

Oracle ODBC ドライバー バージョン 19 はサポート対象外です。

Silk Test Workbench 言語リファレンス

このセクションでは、Silk Test Workbench がサポートする各テクノロジー ドメインのクラス、メソッド、およびプロパティを示します。

Silk Test Workbench のスクリプト言語は Visual Basic .NET です。

Visual Basic .NET のネイティブのクラス、メソッド、プロパティの詳細については、<http://msdn.microsoft.com/ja-jp/library/2x7h1hfk> を参照してください。

Common Class Reference

Lists the available classes for testing Win32, Java SWT, and Windows Forms controls.

BaseGuiTestObject Class

Description

The base class for GUI objects that can have a context menu.

Inheritance Hierarchy

- [GuiTestObject](#)
 - BaseGuiTestObject
 - [AccessibleControl](#)
 - [Control](#)
 - [Window](#)

Syntax

```
'Declaration  
Public Class BaseGuiTestObject _  
Inherits GuiTestObject _  
Implements IScrollable
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)

Name	Description
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

CharSet Enumeration

Description

Specifies the character set to use when reading files.

Inheritance Hierarchy

- [SystemFunctions](#)
 - CharSet

Syntax

```
Public Enumeration CharSet {
    Undefined = 100,
    ANSI = 101,
    UTF8 = 102,
    Unicode = 103
}
```

Members

Name	Description
Undefined	No character set is specified.
ANSI	The ANSI character set is used.
UTF8	The UTF8 character set is used.
Unicode	The Unicode character set is used.

CheckBox Class

Description

The class for check box controls.

Inheritance Hierarchy

- [Control](#)
 - CheckBox

Syntax

```
'Declaration  
Public Class CheckBox _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
State	The state of a check box. Values include: 1=checked, 2=unchecked, 3=undecided.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box.

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsChecked	Returns <i>true</i> if the checkbox is checked and <i>false</i> otherwise.
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects a check box.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetUndecided	Sets the check box to the undecided state.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
Toggle	Toggles the state of the check box. If the checkbox is checked it is unchecked and vice versa. If the checkbox is in the undecided state it is left undecided.

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks the check box.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

CheckBoxToolItem Class

Description

The class for a check box element in a toolbar control.

Inheritance Hierarchy

- [ToolItem](#)
 - CheckBoxToolItem

Syntax

```
'Declaration  
Public Class CheckBoxToolItem _  
Inherits ToolItem
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
State	The state of a check box. Values include: 1=checked, 2=unchecked,3=undecided
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForObject</i>	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from <i>TestObject</i>)
<i>WaitForProperty</i>	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForPropertyNotEquals</i>	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForScreenshotStable</i>	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

ComboBox Class

Description

The class for controls that have both a popup list and a text field. If the user selects an item from the list, the text field is filled with that string. Alternatively, the user can type the string into the text field.

Inheritance Hierarchy

- [Control](#)
 - ComboBox

Syntax

```
'Declaration  
Public Class ComboBox _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the combo box list.
Items	A list of items in the combo box list.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	The index of the selected item.
SelectedItem	The selected item.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ClearText	Removes all text from the combo box.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item from the combo box.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetText	Replaces the text in the text field of the combo box.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Control Class

Description

The class for any control in a window.

Inheritance Hierarchy

- [BaseGuiTestObject](#)
 - Control
 - [CBanner](#)
 - [CheckBox](#)
 - [ComboBox](#)
 - [CoolBar](#)
 - [CTabFolder](#)
 - [DataGrid](#)
 - [DomainUpDown](#)
 - [ElementHost](#)
 - [ExpandBar](#)
 - [FormsHost](#)
 - [Group](#)
 - [Header](#)
 - [Label](#)
 - [Link](#)
 - [ListBox](#)
 - [ListView](#)
 - [MenuStrip](#)
 - [MonthCalendar](#)
 - [Pager](#)
 - [ProgressBar](#)
 - [PushButton](#)
 - [RadioList](#)
 - [Sash](#)
 - [SashForm](#)
 - [Scale](#)
 - [ScrollableControl](#)
 - [ScrollBar](#)
 - [ScrolledComposite](#)
 - [Spinner](#)
 - [StatusBar](#)
 - [SWTBrowser](#)
 - [SWTDateTime](#)
 - [TabControl](#)
 - [Table](#)
 - [TextField](#)
 - [ToggleButton](#)
 - [ToolBar](#)
 - [Tree](#)

- [UpDown](#)
- [ViewForm](#)

Syntax

```
'Declaration
Public Class Control _
Inherits BaseGuiTestObject
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is

Name	Description
	passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
WaitForObject	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

SystemFunctions Class

Description

Allows clients to perform operations on the remote Open Agent host.

Inheritance Hierarchy

- SystemFunctions
 - [CharSet](#)
 - [ExecutionMode](#)
 - [ExecutionResult](#)
 - [FileHandle](#)
 - [FileInfo](#)

- [FileOpenMode](#)
- [FilePointerMode](#)
- [FileShareMode](#)
- [FileSizeUnit](#)
- [RegistryCategory](#)
- [RegistryView](#)

Syntax

'Declaration

Public Class SystemFunctions _

Properties

Name	Description
AgentDirectory	Gets the directory from which the Open Agent is running.
AgentVersion	Gets the version of the Open Agent.
ClipboardText	Gets or sets the text in the clipboard.
CurrentDrive	Gets or sets the current drive of the machine on which the Open Agent is hosted.
CurrentRegistryView	Gets or sets the current registry view.
CurrentWorkingDirectory	Gets or sets the current working directory of the Open Agent.
CursorPosition	Gets the current cursor position of the machine on which the Open Agent is running.
CursorType	Gets the current cursor type of the machine on which the Open Agent is running.
FreeMemory	Gets the percentage of free memory that is available on the machine on which the Open Agent is hosted.
HostName	Gets the hostname of the machine on which the Open Agent is hosted.
Locale	Gets the locale of the machine on which the Open Agent is hosted, for example en-US.
OSName	Gets the name of the operating system of the machine on which the Open Agent is hosted.
OSVersionType	Gets the version type, server or client, of the operating system of the machine on which the Open Agent is hosted.
OSVersion	Gets the major and minor version of the operating system of the machine on which the Open Agent is hosted.

Methods

Name	Description
CompareBinaryFiles	Performs a binary comparison on two files.
CompareBitmaps	Compares two bitmaps.
CompareTextFiles	Compares two text files lexicographically.

Name	Description
<i>CopyFile</i>	Copies a file.
<i>CreateDirectory</i>	Creates a new directory with the specified name.
<i>CreateRegistryKey</i>	Creates a new key in the registry. Recursively creates any missing registry keys in the given keyPath.
<i>CreateRegistryValue</i>	Creates a new value.
<i>DeleteRegistryKey</i>	Deletes a key from the registry.
<i>DeleteRegistryValue</i>	Deletes a value.
<i>DirectoryExists</i>	Checks whether the specified directory exists.
<i>Execute</i>	Executes the specified command on the machine on which the Open Agent is hosted.
<i>ExistsRegistryKey</i>	Checks whether the specified key exists in the registry.
<i>ExistsRegistryValue</i>	Checks if a value exists in the registry.
<i>FileClose</i>	Closes a previously opened file.
<i>FileExists</i>	Checks whether the specified file exists.
<i>FileOpen</i>	Opens a file. If the file does not exist, it is created.
<i>FileReadLine</i>	Reads a single line from a file. Starts at the current position of the pointer in the file.
<i>FileSetPointer</i>	Sets the position pointer within the file for read and write operations.
<i>FileWriteLine</i>	Writes a line at the current position of the pointer in a file.
<i>GetBitmapCRC</i>	Returns the directory from which the Open Agent is running.
<i>GetDirectoryContents</i>	Gets the contents of a directory.
<i>GetEnvironmentVariable</i>	Gets the value of the specified environment variable.
<i>GetFileInfo</i>	Gets information about the specified file.
<i>GetFreeDiskSpace</i>	Returns how much free disk space is available on the specified drive.
<i>GetRegistryKeyNames</i>	Returns the names of all registry keys below the specified key.
<i>GetRegistryValueNames</i>	Returns the values of all registry keys below the specified key.
<i>GetRegistryValue</i>	Returns the value of the specified registry key.
<i>MoveFile</i>	Moves a file.
<i>RemoveDirectory</i>	Deletes the specified directory.
<i>RemoveFile</i>	Deletes the specified file.
<i>SetEnvironmentVariable</i>	Sets the value of the specified environment variable.
<i>SetRegistryValue</i>	Sets the value of the specified registry key.

Desktop Class

Description

Desktop is the class for the entire screen.

Inheritance Hierarchy

- [TestObject](#)
 - Desktop

Properties

Desktop inherits all its properties from [TestObject](#).

Methods

Desktop inherits all its methods from [TestObject](#).

Dialog Class

Description

The class for dialogs.

Inheritance Hierarchy

- [Window](#)
 - Dialog

Syntax

```
'Declaration  
Public Class Dialog _  
Inherits Window
```

Properties

Name	Description
Application	The name of the Application that this Window belongs to. (Inherited from Window)
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)

Name	Description
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)

Name	Description
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DropDownToolItem Class

Description

The class for drop-down element in a toolbar control.

Inheritance Hierarchy

- [ToolItem](#)
 - DropDownToolItem

Syntax

```
'Declaration  
Public Class DropDownToolItem _  
Inherits ToolItem
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Open	Opens an item.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

ExecutionMode Enumeration

Description

Defines whether a function waits for the call to finish before executing a command.

Inheritance Hierarchy

- [SystemFunctions](#)
 - ExecutionMode

Syntax

```
Public Enum ExecutionMode
```

Members

Name	Description
WaitUntilFinished	Execute should return only after the program is finished.
ContinueRunning	Start the executable and immediately return. ExecutionResult.ExitCode will always be 0, and no output will be captured in ExecutionResult.RawOutput.

ExecutionResult Class

Description

Encapsulates the results of a call to SystemFunctions.Execute.

Inheritance Hierarchy

- [SystemFunctions](#)
 - ExecutionResult

Methods

Name	Description
GetExitCode	Returns the result of a call to Execute . 0 if no error occurred.
GetRawOutput	Returns the contents of stdout and stderr.
GetOutput	Returns the contents of stdout and stderr as a string.
ToString	Returns the exit code as a string.

FileHandle Class

Description

A logical file handle.

Inheritance Hierarchy

- [SystemFunctions](#)
 - FileHandle

Methods

Name	Description
GetHandle	Returns the internal value for the handle.
GetName	Returns the name of the file or directory to which the handle corresponds.

FileInfo Class

Description

Structure storing information about a directory entry.

Inheritance Hierarchy

- [SystemFunctions](#)
 - FileInfo

Properties

Name	Description
Attributes	Gets the attributes of the file.
CreationDate	Gets the date at which the file or directory was created.
IsDirectory	Gets whether the instance stores information about a directory or not.
ModificationTime	Gets the time at which the file or directory was last modified.
Name	Gets the name of the file.
Size	Gets the size of the file.

FileOpenMode Enumeration

Description

Defines how to open a file. The file mode is a required parameter for the [FileOpen](#) function.

Inheritance Hierarchy

- [SystemFunctions](#)
 - FileOpenMode

Syntax

```
Public Enum FileOpenMode
```

Members

Name	Description
Read	Opens the file for reading.
Write	Opens the file for writing. If the file does not exist, it is created. If the file exists, it is truncated to zero (0) bytes.
ReadWrite	Opens the file for reading and writing.
Update	Opens the file for writing, but does not truncate the file. Denies write access to other users by default. The file pointer is positioned at the beginning of the file.

FilePointerMode Enumeration

Description

Specifies the position to use when reading or writing a file.

Inheritance Hierarchy

- [SystemFunctions](#)
 - FilePointerMode

Syntax

```
Public Enum FilePointerMode
```

Members

Name	Description
Start	Sets the pointer to the start of the file.
Relative	Leaves the pointer in the current position in the file.
End	Sets the pointer to the end of the file.

FileShareMode Enumeration

Description

Defines how other programs can access files while they are opened through Silk Test.

Inheritance Hierarchy

- [SystemFunctions](#)
 - FileShareMode

Syntax

```
Public Enum FileShareMode
```

Members

Name	Description
DenyNone	Other programs can read from and write to the file at the same time as the Open Agent.
DenyWrite	Other programs can read from but not write to the file at the same time as the Open Agent.
DenyAll	Other programs cannot access the file at the same time as the Open Agent.

FileSizeUnit Enumeration

Description

Defines which unit to use when calculating file sizes.

Inheritance Hierarchy

- [SystemFunctions](#)
 - FileSizeUnit

Syntax

```
Public Enum FileSizeUnit
```

Members

Name	Description
Byte	1 Byte.
KiloByte	2 ¹⁰ Byte.
MegaByte	2 ²⁰ Byte.
GigaByte	2 ³⁰ Byte.
TeraByte	2 ⁴⁰ Byte.

Group Class

Description

The class used to combine controls to give the user visual hints that those controls belong together. Consequently, the Group control does not have any other functionality that can be expressed in methods.

Inheritance Hierarchy

- [Control](#)
 - Group

Syntax

```
'Declaration  
Public Class Group _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

GuiTestObject Class

Description

The base class for all GUI objects.

Inheritance Hierarchy

- [TestObject](#)
 - GuiTestObject
 - [BaseGuiTestObject](#)
 - [Item](#)
 - [Menu](#)

Syntax

```
'Declaration
Public Class GuiTestObject _
Inherits TestObject _
Implements IClickable, IFocusable, IKeyable, INativeWindow
```

Properties

Name	Description
Background	The background color of the GUI object.
Enabled	Whether the GUI object is enabled.
Font	The font type of the GUI object.
Foreground	The foreground color of the GUI object.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

HorizontalScrollBar Class

Description

The class for horizontal scroll bars.

Inheritance Hierarchy

- [ScrollBar](#)
 - [HorizontalScrollBar](#)

Syntax

```
'Declaration  
Public Class HorizontalScrollBar _  
Inherits ScrollBar
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	The internal page size of the scroll bar. (Inherited from ScrollBar)
Position	The actual position of the scroll bar caret. (Inherited from ScrollBar)
Range	The range of the scroll bar. (Inherited from ScrollBar)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PageLeft	Scrolls the scroll bar page-wise left.
PageRight	Scrolls the scroll bar page-wise right.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Scrolls the scroll bar. (Inherited from ScrollBar)
ScrollLeft	Scrolls the scroll bar to the left.
ScrollRight	Scrolls the scroll bar to the right.
ScrollToMax	Scrolls the scroll bar to its maximum position. (Inherited from ScrollBar)
ScrollToMin	Scrolls the scroll bar to its minimum position. (Inherited from ScrollBar)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

IClickable Interface

Description

Interface for objects that can be clicked.

Syntax

```
'Declaration
Public Class IClickable
```

Methods

Name	Description
Click	Clicks on the object.
DoubleClick	Double-clicks a mouse button on the object.
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications.
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications.
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications.

IFocusable Interface

Description

Interface for objects that can be focused on.

Syntax

```
'Declaration
Public Class IFocusable
```

Methods

Name	Description
IsFocused	Return whether the control has focus.
SetFocus	Gives focus to the control.

IKeyable Interface

Description

Interface for objects that use keystrokes. All methods and properties in this interface are not supported for mobile Web applications.

Syntax

```
'Declaration  
Public Class IKeyable
```

Methods

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons.
ReleaseKeys	Releases a set of keys or mouse buttons.
TypeKeys	Sends a set of keystrokes to the object.
TypePasswordKeys	Types an encrypted password into an object, for example a text field.

IMoveable Interface

Description

Interface for objects that can be moved, e.g. windows.

Syntax

```
'Declaration  
Public Class IMoveable
```

Properties

Name	Description
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored

Methods

Name	Description
Close	Closes the window.
CloseSynchron	Closes the window and waits until the window is closed.
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open.
GetFocus	Returns the object with the input focus.
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window.
IsActive	Returns whether the window is set to active.
Maximize	Maximizes the window.
Minimize	Reduces the window to an icon.
Move	Moves the window.
Restore	Restores the window to its previous size.
SetActive	Makes the window active.

Name	Description
Size	Resizes the window.

INativeWindow Interface

Description

Interface for objects that provide a native window handle.

Syntax

```
'Declaration
Public Class INativeWindow
```

Properties

Name	Description
NativeHandle	The native window handle for the object.

IScrollable Interface

Description

Interface for objects that can have scroll bars.

Syntax

```
'Declaration
Public Class IScrollable
```

Methods

Name	Description
GetHorizontalBar	Returns the horizontal scroll bar for this control.
GetVerticalBar	Returns the vertical scroll bar for this control.

Item Class

Description

Item base class.

Inheritance Hierarchy

- [GuiTestObject](#)
 - Item
 - [CoolItem](#)
 - [CTabItem](#)
 - [DataGridColumn](#)
 - [DataGridItem](#)
 - [DataGridRow](#)

- [ExpandItem](#)
- [MenuItem](#)
- [SeparatorItem](#)
- [SWTTabItem](#)
- [SWTTreeColumn](#)
- [TableColumn](#)
- [TableRow](#)
- [ToolItem](#)

Syntax

```
'Declaration
Public Class Item _
Inherits GuiTestObject
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
WaitForObject	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

Label Class

Description

The class for static text strings (that is, text that the user cannot change, such as a label).

Inheritance Hierarchy

- [Control](#)
 - Label

Syntax

```
'Declaration
Public Class Label _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT

Name	Description
	takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Link Class

Description

The class that opens the topic location referenced in the source.

Inheritance Hierarchy

- [Control](#)
 - Link

Syntax

```
'Declaration
Public Class Link _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)

Name	Description
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ListBox Class

Description

The class for list boxes.

Inheritance Hierarchy

- [Control](#)
 - ListBox
 - [CheckedListBox](#)

Syntax

```
'Declaration
Public Class ListBox _
Inherits Control
```

Properties

Name	Description
AllowsMultiSelect	Whether the control supports selecting multiple items.
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)

Name	Description
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the list box.
Items	A list of items in the list box.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	The index of the first selected item.
SelectedIndices	The indices of the selected item(s).
SelectedItem	The name of the first selected item.
SelectedItems	The names of the selected item(s).
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleSelect	Double-clicks an item in the list box.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>ExtendSelect</i>	Selects a range of items by extending the selection in the list box.
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetHorizontalBar</i>	Returns the horizontal scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MultiSelect</i>	Selects an item in the multi- or extend-selection list box.
<i>MultiUnselect</i>	Unselects an item in the multi- or extend-selection list box.
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Select</i>	Selects an item from the list box.
<i>SelectRange</i>	Selects a range of items in the extend-selection list box.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Menu Class

Description

The class for menus.

Inheritance Hierarchy

- [GuiTestObject](#)
 - Menu

Syntax

```
'Declaration  
Public Class Menu _  
Inherits GuiTestObject
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemCount	Returns the number of menu items for this menu. Also includes separator menu items.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects a menu.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

MenuItem Class

Description

The class for items on a menu.

Inheritance Hierarchy

- [Item](#)
 - MenuItem

Syntax

```
'Declaration
Public Class MenuItem _
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Checked	Whether the menu item is checked.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item from the menu.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

PushButton Class

Description

The class for buttons.

Inheritance Hierarchy

- [Control](#)
 - PushButton

Syntax

```
'Declaration
Public Class PushButton _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)

Name	Description
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Pressed	Whether the pushbutton is pressed.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetHorizontalBar</i>	Returns the horizontal scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects a pushbutton.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

PushToolItem Class

Description

The class for a pushbutton element in a toolbar control.

Inheritance Hierarchy

- [ToolItem](#)
 - PushToolItem

Syntax

```
'Declaration
Public Class PushToolItem _
Inherits ToolItem
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)

Name	Description
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item.

Name	Description
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

RadioList Class

Description

The class for a group of radio buttons, such as the two radio buttons used to specify direction in the Find dialog of the Text Editor application. Only one button in a radio list can be selected at a single time.

Inheritance Hierarchy

- [Control](#)
 - RadioList

Syntax

```
'Declaration
Public Class RadioList _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the radio list.
Items	A list of items in the radio list.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)

Name	Description
SelectedIndex	The index of the selected item.
SelectedItem	The item to select.
State	The state of a check box. Values include: 1=checked, 2=unchecked,3=undecided
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetHorizontalBar</i>	Returns the horizontal scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item from the radiolist.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

RadioListItem Class

Description

RadioListItem is the class for a radio button element in a toolbar control.

Inheritance Hierarchy

- [ToolItem](#)
 - RadioListItem

Syntax

```
'Declaration
Public Class RadioListItem _
Inherits ToolItem
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)

Name	Description
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
State	The state of a check box. Values include: 1=checked, 2=unchecked,3=undecided
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Select</i>	Selects an item.

Name	Description
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

RegistryCategory Enumeration

Description

Specifies the root key for the registry. For additional information, see [Predefined Keys](#).

Inheritance Hierarchy

- [SystemFunctions](#)
 - RegistryCategory

Syntax

```
Public Enum RegistryCategory
```

Members

Name	Description
HKEY_CLASSES_ROOT	Registry entries subordinate to this key define types (or classes) of documents and the properties associated with those types.
HKEY_CURRENT_USER	Registry entries subordinate to this key define the preferences of the current user.
HKEY_LOCAL_MACHINE	Registry entries subordinate to this key define the physical state of the computer.
HKEY_USERS	Registry entries subordinate to this key define the default user configuration for new users on the local computer and the user configuration for the current user.

RegistryView Enumeration

Description

Defines whether to use the 32bit or the 64bit registry.

Inheritance Hierarchy

- [SystemFunctions](#)

- [RegistryView](#)

Syntax

```
Public Enum RegistryView
```

Members

Name	Description
RegistryView_32Bit	Use the 32bit registry.
RegistryView_64Bit	Use the 64bit registry.

Scale Class

Description

The class for scales.

Inheritance Hierarchy

- [Control](#)
 - Scale

Syntax

```
'Declaration
Public Class Scale _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Position	The actual position of the scale caret.
Range	The range of the scale.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need

Name	Description
	this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetPosition	Sets the position of the scale.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetToMax	Sets the scale to its maximum position.

Name	Description
SetToMin	Sets the scale to its minimum position.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter,

Name	Description
WaitForScreenshotStable	<p>the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

ScrollBar Class

Description

The class for scroll bars, including the scroll bars that are parts of controls, such as the scroll bar on a list box.

Inheritance Hierarchy

- [Control](#)
 - ScrollBar
 - [HorizontalScrollBar](#)
 - [VerticalScrollBar](#)

Syntax

```
'Declaration
Public Class ScrollBar _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	The internal page size of the scroll bar.
Position	The actual position of the scroll bar caret.
Range	The range of the scroll bar.

Name	Description
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Scroll	Scrolls the scroll bar.
ScrollToMax	Scrolls the scroll bar to its maximum position.
ScrollToMin	Scrolls the scroll bar to its minimum position.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout

Name	Description
<i>WaitForPropertyNotEquals</i>	using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
<i>WaitForScreenshotStable</i>	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SeparatorItem Class

Description

The class for menu separators.

Inheritance Hierarchy

- [Item](#)
 - SeparatorItem

Syntax

```
'Declaration
Public Class SeparatorItem _
Inherits Item
```

Properties

Name	Description
<i>Background</i>	The background color of the GUI object. (Inherited from GuiTestObject)
<i>Enabled</i>	Whether the GUI object is enabled. (Inherited from GuiTestObject)
<i>Font</i>	The font type of the GUI object. (Inherited from GuiTestObject)
<i>Foreground</i>	The foreground color of the GUI object. (Inherited from GuiTestObject)
<i>NativeHandle</i>	The native window handle for the object. (Inherited from INativeWindow)

Name	Description
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

TabControl Class

Description

The class for tabbed, multi-page dialogs and for button bars.

Inheritance Hierarchy

- [Control](#)
 - TabControl
 - [SWTTabControl](#)

Syntax

```
'Declaration
Public Class TabControl _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the tab control.
Items	A list of items in the tab control.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	The index of the selected item.
SelectedItem	The selected item.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Sets the current page to the specified page.
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForPropertyNotEquals 	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Table Class

Description

The class for tables.

Inheritance Hierarchy

- [Control](#)
 - Table
 - [SWTTable](#)

Syntax

```
'Declaration
Public Class Table _
Inherits Control
```

Properties

Name	Description
AllowsCheck	Whether the control can display a checkmark.
AllowsMultiSelect	Whether the table supports selecting multiple items.
Background	The background color of the GUI object. (Inherited from GuiTestObject)
ColumnCount	The number of columns in the table.
ColumnItems	A list of all items in the column.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)

Name	Description
RowCount	The number of rows in the table.
RowItems	A list of all table rows in the table.
SelectedIndices	The indices of the selected item(s).
SelectedItems	The selected item(s).
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box in the defined row of a table.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Double-clicks an item.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
ExtendSelectRow	Selects a range of rows.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)

Name	Description
FocusRow	Focuses on a row in the table.
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MultiSelectRow	Adds a row from the table to the set of selected rows.

Name	Description
MultiUnselectRow	Removes a row from the set of selected rows.
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SelectRow	Selects an row in the table.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks the check box in the defined row of a table.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

TableColumn Class

Description

The class for columns in a table.

Inheritance Hierarchy

- [Item](#)
 - TableColumn
 - [SWTTableColumn](#)

Syntax

```
'Declaration
Public Class TableColumn _
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
Width	The width of the column.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Resize	Resizes a column.
Select	Selects a column.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By

Name	Description
	default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

TableRow Class

Description

The class for rows in a table.

Inheritance Hierarchy

- [Item](#)
 - TableRow
 - [SWTTableRow](#)

Syntax

```
'Declaration
Public Class TableRow _
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Checked	Whether the check box in the row is checked.

Name	Description
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the row.
Items	A list of items in the row.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

TestObject Class

Description

The base class of all UI objects.

Syntax

```
'Declaration
Public Class TestObject
```

Properties

Name	Description
Text	The text of the control.
Value	The value of the control, e.g.: text in a text control.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose.

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately.
Exists	Checks if an object exists in the application under test.
Find	Finds an object specified by an XPath locator.
FindAll	Finds all objects specified by an XPath locator.
GenerateLocator	Returns a locator for this object.
GetChildren	Returns the child objects of this object.
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject.
GetParent	Looks up the parent of this object in the test application.
GetProperty	Returns the value of the specified property.
GetPropertyList	Returns a list of property names that can be retrieved for the given object.
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object.
HighlightObject	Highlights this object.
ImageClick	Clicks on specified image asset.
ImageClickFile	Clicks on the specified image.
ImageExists	Returns whether the specified image asset exists.
ImageExistsFile	Returns whether the specified image exists.
ImageRectangle	Returns the object-relative rectangle of the specified image asset.
ImageRectangleFile	Returns the object-relative rectangle of the specified image.
Invoke	Dynamically invokes a method on the test object.
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject.
SetProperty	Sets the value of the specified property.
StartScreenRecording	Starts screen recording
StopScreenRecording	Stops screen recording
TextCapture	Returns the text in this object's visible area.
TextClick	Clicks in the center of the specified text.
TextExists	Returns whether the specified text exists.
TextRectangle	Returns the object-relative rectangle of the specified text.

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached.
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached.
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts.
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached.
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached.
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached.

TextField Class

Description

The class for single-line and multi-line fields whose text can be modified by the user. The class also supports the Windows RichEdit control.

Inheritance Hierarchy

- [Control](#)

- `TextField`
 - [StyledText](#)

Syntax

```
'Declaration
Public Class TextField _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
IsPassword	Whether the control is a password text field.
MultiLine	Whether the control is multiline.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Position	The position of the caret within the text field.
SelectedRange	The selected range within the text field.
SelectedText	The selected text within the text field. An empty string if no text is selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is <code>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps</code> . The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ClearText	Removes all text from the text field.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetMultiText	Returns the specified lines of text in the multi-line text field.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetMultiText</i>	Substitutes all or part of the lines in the multi-line text field.
<i>SetPosition</i>	Sets the insertion point in the text field.
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>SetSelectionRange</i>	Selects the specified range of the single-line or multi-line text field.
<i>SetText</i>	Substitutes new text for all or part of the text in the text field.
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ToggleButton Class

Description

The class for buttons that have a state that can be changed by selecting the button.

Inheritance Hierarchy

- [Control](#)
 - ToggleButton

Syntax

```
'Declaration
Public Class ToggleButton _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
State	The state of a toggle button. Values include: 1=checked, 2=unchecked, 3=undecided.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ToolBar Class

Description

The class for tool bar controls. A tool bar is a container for a set of buttons and other standard controls.

Inheritance Hierarchy

- [Control](#)
 - ToolBar

Syntax

```
'Declaration
Public Class ToolBar _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ToolItem Class

Description

The class for a tool item in a toolbar control.

Inheritance Hierarchy

- [Item](#)
 - ToolItem
 - [CheckBoxToolItem](#)
 - [DropDownToolItem](#)
 - [PushToolItem](#)
 - [RadioListToolItem](#)

Syntax

```
'Declaration
Public Class ToolItem _
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Tree Class

Description

The class for items organized into graphical hierarchies.

Inheritance Hierarchy

- [Control](#)
 - Tree
 - [SWTTree](#)

Syntax

```
'Declaration  
Public Class Tree _  
Inherits Control
```

Properties

Name	Description
AllowsCheck	Whether the control can display a checkmark.
AllowsMultiSelect	Whether the control supports selecting multiple items.
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the tree (including all children).
ItemPaths	A list of items in the tree (including all children).
Items	A list of items in the tree (including all children).
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	The index of the first selected item.
SelectedIndices	The indices of the selected item(s).
SelectedItem	The name of the first selected item.
SelectedItems	The names of the selected item(s).
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
VisibleItemPaths	A list of visible items in the tree (including all children).
VisibleItems	A list of visible items in the tree (including all children).

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box.
Click	Clicks on the object. (Inherited from IClickable)
Collapse	Collapses an item in a treeview control.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Double-clicks an item.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Expand	Expands an object in a treeview control.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetItemPaths	Returns all items of the tree as a list of itempaths
GetItemRect	Returns the size and position of an item relative to the treeview control.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsExpandable	Checks if the given item can be expanded.
IsExpanded	Checks if the given item is expanded.
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item from the tree.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks the check box.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

VerticalScrollBar Class

Description

The class for vertical scroll bars.

Inheritance Hierarchy

- [ScrollBar](#)
 - VerticalScrollBar

Syntax

```
'Declaration
Public Class VerticalScrollBar _
Inherits ScrollBar
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	The internal page size of the scroll bar. (Inherited from ScrollBar)
Position	The actual position of the scroll bar caret. (Inherited from ScrollBar)
Range	The range of the scroll bar. (Inherited from ScrollBar)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PageDown	Scrolls the scroll bar page-wise down.

Name	Description
PageUp	Scrolls the scroll bar page-wise up.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Scrolls the scroll bar. (Inherited from ScrollBar)
ScrollDown	Moves the scroll bar down.
ScrollToMax	Scrolls the scroll bar to its maximum position. (Inherited from ScrollBar)
ScrollToMin	Scrolls the scroll bar to its minimum position. (Inherited from ScrollBar)
ScrollUp	Moves the scroll bar up.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
WaitForObject	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

Window Class

Description

The class for windows that typically can move.

Inheritance Hierarchy

- [BaseGuiTestObject](#)
 - Window
 - [BrowserApplication](#)
 - [Dialog](#)
 - [FlexStandalonePlayer](#)
 - [FormsWindow](#)

- [Shell](#)
- [UIAWindow](#)

Syntax

```
'Declaration
Public Class Window _
Inherits BaseGuiTestObject _
Implements IMoveable
```

Properties

Name	Description
Application	The name of the Application that this Window belongs to.
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 Size 	Resizes the window. (Inherited from IMoveable)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

コア クラス リファレンス

アプリケーションを設定すると、Silk Test Workbench は標準のエージェントおよびテスト オブジェクト コントロールのテストの組み込みサポートを自動的に提供します。

ActiveData クラス

説明

ActiveData は、ActiveData オブジェクトをスクリプトで使用するための機能を提供するクラスです。このクラスは内部コンストラクターです。ActiveData を呼び出すには、このクラスを使用する代わりに、Workbench.LoadActiveData を呼び出します。

継承

ActiveData はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration
Public Class ActiveData
```

プロパティ

名前	説明
ColumnCount	基になる ActiveData オブジェクトの列の数。
Count	ActiveData 行の列の数。

メソッド

名前	説明
AddColumn	基になる ActiveData オブジェクトに列を追加します。
AddRow	基になる ActiveData オブジェクトに行を追加します。
Dispose	オブジェクトで使用されるリソースをクリーン アップします。

名前	説明
GetEnumerator	行の列挙子を返します。
Item	ActiveDataRow をインデックス別に返します。
RemoveColumn	基になる ActiveData オブジェクトから列を削除します。
RemoveRow	基になる ActiveData オブジェクトから行を削除します。
RenameColumn	ActiveData オブジェクトの列の名前を変更します。
Reset	フォーカスを ActiveData ファイルの先頭行に移動します。
Save	ActiveData オブジェクトを保存します。

AddColumn メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

基になる ActiveData オブジェクトに列を追加し、成功または失敗を示す値を返します。

構文

AddColumn (defaultValue, columnName)

変数	説明
defaultValue	新しい列のデフォルト値。 Double。
columnName	列の名前。 String。

使用例

PhoneBookData 資産に「Occupation」という新しい列を追加するには、以下のように入力します。

```
Public Sub Main()
    AddColumnToDataFile( "occupation" )
End Sub

Public Sub AddColumnToDataFile( name As String )
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData" )
    data.AddColumn( "Occupation", name )
    data.Save()
End Sub
End Module
```

AddRow メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

基になる ActiveData オブジェクトに行を追加し、成功または失敗を示す値を返します。

構文

```
AddRow ()
```

使用例

特定の ID 番号の新しいデータ行を追加するには、以下のように入力します。

```
Public Sub Main()  
    InsertNewNumberToDataFileWithOccupation( 108, "Jay", "Jones", "5551219", _  
        "QA Analyst" )  
End Sub  
  
Public Sub InsertNewNumberToDataFileWithOccupation( id As Integer, _  
    firstName As String, lastName As String, phoneNumber As String, _  
    occupation As String )  
    Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData" )  
    Dim row As ActiveDataRow = data.AddRow()  
  
    row.SetLong( "id", id )  
    row.SetString( "fname", firstName )  
    row.SetString( "lname", lastName )  
    row.SetString( "number", phoneNumber )  
    row.SetString( "occupation", occupation )  
  
    data.Save()  
End Sub  
End Module
```

Dispose メソッド (ActiveData)

クラス

[ActiveData](#)。

処理


オブジェクトで使用されるリソースをクリーン アップします。

構文

```
Dispose ()
```

メモ

このメソッドは、複数のスクリプトからファイルを読み書きしている場合のみ使用する必要があります。

 **注意:** 上記以外のどの場合でも、このメソッドは使用しないでください。Silk Test Workbench は必要に応じてこのメソッドを自動的に呼び出します。

例

```
Dim a As ActiveData  
a = Workbench.LoadActiveData("ActiveDataAsset")  
' Do your work here  
a.Dispose()
```


GetEnumerator メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

行の列挙子を返します。

構文

```
GetEnumerator ()
```

Item メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

ActiveDataRow をインデックス別に返します。

構文

```
Item(index)
```

変数	説明
<i>index</i>	返す行のインデックス。Integer。

RemoveRow メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

基になる ActiveData オブジェクトから行を削除します。

構文

```
RemoveRow (rowNumber)
```

変数	説明
<i>rowNumber</i>	削除する行の数 (1 から始まる)。Integer。

RemoveColumn メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

基になる ActiveData オブジェクトから列を削除し、成功または失敗を示す値を返します。

構文

RemoveColumn (*columnName*)

変数	説明
<i>columnName</i>	削除する列の名前 (1 から始まる)。Integer。

RenameColumn メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

ActiveData オブジェクトの列の名前を変更し、列の名前が正常に変更されたかどうかを示す値を返します。

構文

RenameColumn (*columnIndex*, *newColumnName*)

変数	説明
<i>columnIndex</i>	削除する列のインデックス (1 から始まる)。Integer。
<i>newColumnName</i>	列の新しい名前。String。

Reset メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

フォーカスを ActiveData ファイルの先頭行に移動します。

構文

Reset ()

Save メソッド (ActiveData)

クラス

[ActiveData](#)。

処理

ActiveData オブジェクトを保存します。

構文

Save ()

ColumnCount プロパティ

クラス

[ActiveData](#)。

処理

基になる ActiveData オブジェクトの列の数を返します。

構文

```
'Declaration  
Protected Read Property ColumnCount As Integer
```

アクセス

読み取り専用。

Count プロパティ

クラス

[ActiveData](#)。

処理

ActiveData 行の列の数を返します。

構文

```
'Declaration  
Protected Read Property Count As Integer
```

アクセス

読み取り専用。

ActiveDataRow クラス

説明

ActiveDataRow は、ActiveData オブジェクトの行を使用するための機能を提供するクラスです。このクラスは内部コンストラクターです。

継承

ActiveDataRow はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class ActiveDataRow
```

プロパティ

名前	説明
Count	ActiveData 行の列の数。
ExternalRowNumber	読み込まれた一連の行における現在のオブジェクト行番号を取得します。
RowNumber	ActiveData ファイル内における現在のオブジェクトの行番号を取得します。

メソッド

名前	説明
Dispose	行で使用されるリソースをクリーン アップします。
GetDouble	列の double 値を取得します。
GetLong	列の integer 値を取得します。
GetLongLong	列の long 値を取得します。
GetString	列の string 値を取得します。
SetDouble	列の double 値を設定します。
SetLong	列の integer 値を設定します。
SetLongLong	列の long 値を設定します。
SetString	列の string 値を設定します。

Dispose メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。


処理

行で使用されるリソースをクリーン アップします。

構文

```
Dispose ()
```

メモ

 **注意:** このメソッドは使用しないでください。 Silk Test Workbench は必要に応じてこのメソッドを自動的に呼び出します。

GetDouble メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の double 値を取得します。

構文

GetDouble (*columnIndex*)

または

GetDouble (*columnName*)

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。

GetLong メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の integer 値を取得します。

構文

GetLong (*columnIndex*)

または

GetLong (*columnName*)

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。

GetLongLong メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の long 値を取得します。

構文

GetLongLong (*columnIndex*)

または

GetLongLong (*columnName*)

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。

GetString メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の string 値を取得します。

構文

```
GetString (columnIndex)
```

または

```
GetString (columnName)
```

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。

SetDouble メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の double 値を設定します。

構文

```
SetDouble (columnIndex, value)
```

または

```
SetDouble (columnName, value)
```

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。
<i>value</i>	新しい値。Integer または Double。

SetLong メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の integer 値を設定します。

構文

SetLong (*columnIndex*, *value*)

または

SetLong (*columnName*, *value*)

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。
<i>value</i>	新しい値。Integer。

SetLongLong メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の long 値を設定します。

構文

SetLongLong (*columnIndex*, *value*)

または

SetLongLong (*columnName*, *value*)

変数	説明
<i>columnIndex</i>	列のインデックス (1 から開始)。Integer。
<i>columnName</i>	列の名前。String。
<i>value</i>	新しい値。Long。

SetString メソッド (ActiveDataRow)

クラス

[ActiveDataRow](#)。

処理

列の string 値を設定します。

構文

SetString (*columnIndex*, *value*)

または

SetString (*columnName*, *value*)

変数	説明
<code>columnIndex</code>	列のインデックス (1 から開始)。Integer。
<code>columnName</code>	列の名前。String。
<code>value</code>	新しい値。String。

Count プロパティ

クラス

[ActiveData](#)。

処理

ActiveData 行の列の数を返します。

構文

```
'Declaration
Protected Read Property Count As Integer
```

アクセス

読み取り専用。

ExternalRowNumber プロパティ

クラス

[ActiveDataRow](#)。

処理

読み込まれた一連の行における現在のオブジェクトの行番号を取得します。

構文

```
'Declaration
Protected Read Property ExternalRowNumber As Integer
```

アクセス

読み取り専用。

RowNumber プロパティ

クラス

[ActiveDataRow](#)。

処理

実際のファイル内の現在のオブジェクトの行番号を取得します。

構文

```
'Declaration
Protected Read Property RowNumber As Integer
```


アクセス

読み取り専用。

Agent クラス

説明

Agent は、グラフィカル ユーザー インターフェイスを操作する Silk Test Workbench のコンポーネントである Open Agent のクラスです。

Agent では識別子 Agent が事前に定義されています。Agent のメソッドを呼び出すには、以下の構文を使用します。

```
Agent.method()
```

継承

Agent はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class Agent
```

プロパティ

名前	説明
Desktop	ルート GUI オブジェクトを取得します。
SystemFunctions	SystemFunctions オブジェクトを取得します。このオブジェクトを使うと、Open Agent がインストールされたローカル マシンやリモート マシン上で、OS 固有の操作を実行できます。

メソッド

名前	説明
Attach	指定したアプリケーションにアタッチします。コマンド ライン パターンおよびテクノロジー ドメインを追加で指定することができます。
Connect	指定したホストとポートを使用して指定したマシン上のエージェントに接続します。
Decrypt	指定された文字列を復号化します。 Encrypt メソッドで暗号化された文字列のみを復号化することができます。
DetachAll	現在アタッチされているすべてのアプリケーションを解除します。
Encrypt	指定された文字列を暗号化します。
ExecuteBaseState	テスト対象アプリケーションが実行中でテストの準備ができていることを保証します。
GetOption	エージェント オプションの値を取得します。
ResetOptions	エージェントのすべてのオプションをデフォルト値に戻します。
SetOption	エージェント オプションを設定します。このオプションは、すべてのテクノロジー ドメインに対して利用されます。

名前	説明
Shutdown	エージェントを停止します。テスト実行が完了したときにのみ呼び出されるべきです。

Attach メソッド

クラス

[Agent](#)。

処理

指定したアプリケーションにアタッチします。コマンド ライン パターンおよびテクノロジー ドメインを追加で指定することができます。

構文

`Attach (executablePattern, commandLinePattern, techDomains)`

変数	説明
<code>executablePattern</code>	アプリケーションの実行可能名。パターンには、ワイルドカード '?' および '*' を使用できます (それぞれ、1 文字、または複数文字に一致)。例: <code>myApplication.exe</code> 、 <code>myApplica?ion.exe</code> 、 <code>myApp*.exe</code> 。String。
<code>commandLinePattern</code>	省略可能: このパターンはアプリケーションのコマンドライン引数に対して照合します。これは、アプリケーションの複数のインスタンスが実行中で、その中のいくつかだけがテスト対象である場合に有効です。Java アプリケーションの場合、特徴的な jar や main クラスの名前をコマンドラインパターンに含めることができます。例: <code>*org.MyMainClass</code> 。特定のテクノロジー ドメインをロードしたいときに、コマンドラインパターンが無いことを指定する場合には、 <code>""</code> を使用できます。String。
<code>techDomains</code>	省略可能: アプリケーションに読み込むテクノロジー ドメイン。通常は、このパラメータはテクニカル サポートによってのみ使用されます。テクノロジー ドメインを指定しないことをお勧めします。テクノロジー ドメインを指定しない場合は、すべての利用可能なテクノロジー ドメインが読み込まれます。TechDomain。

使用例

次の例では、メモ帳にアタッチします。

```
Agent.Attach("notepad.exe")
```

次の例では、アプリケーションの複数のインスタンスが実行されている場合に、コマンドラインパターンを使用してプロセスの特定のインスタンスにアタッチする方法を示します。

```
Agent.Attach("javaw.exe", "*org.MyMainClass")
```

以下の例は、Internet Explorer にアタッチして Win32 テクノロジー ドメインをロードする方法を示しています。

```
Agent.Attach("iexplore.exe", "", {"WIN32"})
```

Connect メソッド

クラス

[Agent](#)。

処理

指定したホストとポートを使用して指定したマシン上のエージェントに接続します。

エージェントが実行されていない場合、またはエージェントがこのマシンに到達できない場合は、例外がスローされます。

指定したホストがローカル ホストの場合は、エージェントが起動します。

このエージェントとの接続を最初に確立した後に、すべてのエージェント オプションをデフォルト値にリセットするには、`Agent.ResetOptions()` を呼び出します。

構文

```
Connect (host, [port])
```

変数	説明
<code>host</code>	Open Agent のホスト。String。
<code>port</code>	省略可能: Open Agent のポート。ポートが指定されていない場合、デフォルトのポートが使用されます。このポートは、Information Service のポートで、デフォルト値は 22901 です。たとえば、別の Information Service のポートでエージェントに接続するには、「 <code>Agent.Connect("myRemoteMachine", portnumber)</code> 」と入力します。 リモートエージェントに接続している場合に、ローカルエージェントに接続するには、「 <code>Agent.Connect("localhost")</code> 」と入力します。Integer。

Decrypt メソッド

クラス

[Agent](#)。

処理

指定された文字列を復号化します。Encrypt メソッドで暗号化された文字列のみ復号化することができます。

構文

```
Decrypt(toDecrypt)
```

変数	説明
<code>toDecrypt</code>	復号化する文字列。String。

DetachAll メソッド

クラス

[Agent](#)。

処理

現在アタッチされているすべてのアプリケーションを解除します。

構文

```
DetachAll( )
```

Encrypt メソッド

クラス

[Agent](#)。

処理

指定された文字列を暗号化します。

構文

```
Encrypt (toEncrypt)
```

変数	説明
<i>toEncrypt</i>	暗号化する文字列。String。

ExecuteBaseState メソッド

クラス

[Agent](#)。

処理

テスト対象アプリケーションが実行中でテストの準備ができていることを保証します。

構文

```
TestObject ExecuteBaseState(baseState)
```

変数	説明
<i>baseState</i>	基本状態を実行するために必要なすべての情報を含みます。IBaseState。

GetOption メソッド

クラス

[Agent](#)。

処理

エージェント オプションの値を取得します。

構文

```
GetOption(name)
```

変数	説明
<i>name</i>	オプションの名前。使用できるオプションのリストは Options クラスで定義されています。String。

例

たとえば、以下のように入力します。

```
desktop.getOption(Options.ObjectResolveTimeout)
```

ResetOptions メソッド

クラス

[Agent](#)。

処理

エージェントのすべてのオプションをデフォルト値に戻します。

構文

```
ResetOptions()
```

SetOption メソッド

クラス

[Agent](#)。

処理

エージェント オプションを設定します。このオプションは、すべてのテクノロジー ドメインに対して利用されます。

構文

```
SetOption(name, value)
```

変数	説明
<i>name</i>	オプションの名前。使用できるオプションのリストは Options クラスで定義されています。String。
<i>value</i>	オプションの値。Object。

例

たとえば、以下のように入力します。

```
Agent.setOption(Options.ObjectResolveTimeout, true)
```

Shutdown メソッド (Agent)

クラス

[Agent](#)。

処理

夜間テスト実行後などのテスト実行後に Open Agent を停止する場合にこのメソッドを呼び出します。このメソッドは、テスト実行の最後にのみ呼び出されるべきで、テストの操作が実行される前に呼び出されるべきではありません。シャットダウン後に操作が実行されると、エージェントは再起動します。Open Agent を停止する方法として、次のコマンドをコマンド ラインから実行する方法もあります：

```
openAgent.exe -shutDown
```

構文

VB

```
Shutdown()
```

例 : Shutdown メソッドの使用

```
// VB code
<AssemblyCleanup(>
Public Shared Sub ShutdownAgentAfterTestRun()
    Agent.Shutdown()
End Sub
```

Desktop プロパティ

クラス

[Agent](#)。

処理

ルート GUI オブジェクトを取得します。

構文

```
'Declaration
Protected Read Property Desktop As Desktop
```

アクセス

読み取り専用。

SystemFunctions プロパティ (Agent)

クラス

[Agent](#)。

処理

SystemFunctions オブジェクトを取得します。このオブジェクトを使うと、Open Agent がインストールされたローカル マシンやリモート マシン上で、OS 固有の操作を実行できます。利用可能なシステム関数についての情報は、「SystemFunctions クラス」トピックを参照してください。

構文

```
'Declaration  
Public Read Property SystemFunctions As SystemFunctions
```

アクセス

読み取り専用。

例

たとえば、メモ帳をリモート マシン上で開始する場合は、次のコードをスクリプトに記述します。

```
Agent.SystemFunctions.Execute("notepad.exe")
```

BaseState クラス

説明

BaseState クラスを使用すると、テスト対象アプリケーションが実行中でテストの準備ができていたことが保証されます。また、テスト対象アプリケーションが前面に配置されます。

エージェントをアプリケーションにアタッチしようとした場合に、BaseState.Locator で指定されたテスト オブジェクトが見つかったら、基本状態としてこのテスト オブジェクトを含むウィンドウが前面に配置され、見つかったテスト オブジェクトが即座に返されます。テスト オブジェクトが見つからない場合、baseStateInfo で指定したとおりにアプリケーションがコマンド ラインから起動し、エージェントがブラウザにアタッチされます。テスト オブジェクトが見つかるまで、エージェントは待機します。待機するためのタイムアウトは、オプション Options.ObjectResolveTimeout.タイムアウトのデフォルト値は、30 秒です。タイムアウトを変更するには、Desktop.SetOption (String, Object) を参照してください。**オプション** ダイアログ ボックスの **再生 > タイミング** にある **アプリケーション準備完了タイムアウト** を編集して UI 上からタイムアウトを変更することもできます。指定したタイムアウト内にテスト オブジェクトが見つかったら、このテスト オブジェクトを含むウィンドウが前面に配置され、見つかったテスト オブジェクトが返されます。テスト オブジェクトが見つからずにタイムアウトした場合は、例外がスローされます。

継承

BaseState はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class BaseState _  
Implements IBaseState
```

プロパティ

名前	説明
CommandLineArguments	コマンド ライン引数を取得または設定します。
CommandLinePattern	コマンド ライン パターンを取得または設定します。
ExecutablePattern	実行可能パターンを取得または設定します。
Executable	実行可能ファイルの名前を取得します
Locator	実行可能ファイルのロケータを返します

名前	説明
TechDomains	テストに使用するテクノロジー ドメインのリストを取得または設定します
WorkingDirectory	作業ディレクトリを取得または設定します。

CommandLineArguments プロパティ

クラス

[BaseState](#)。

処理

コマンド ライン引数を取得または設定します。

構文

```
'Declaration
Public ReadWrite Property CommandLineArguments As String
```

アクセス

読み書き。

CommandLinePattern プロパティ

クラス

[BaseState](#)。

処理

コマンド ライン パターンを取得または設定します。

構文

```
'Declaration
Public ReadWrite Property CommandLinePattern As String
```

アクセス

読み書き。

ExecutablePattern プロパティ

クラス

[BaseState](#)。

処理

実行可能パターンを取得または設定します。

構文

```
'Declaration
Public ReadWrite property ExecutablePattern As String
```


アクセス

読み書き。

Executable プロパティ

クラス

[BaseState](#)。

処理

実行可能ファイルの名前を返します。

構文

```
'Declaration  
Public Read Property CommandLineArguments As String
```

アクセス

読み取り専用。

Locator プロパティ

クラス

[BaseState](#)。

処理

実行可能ファイルのロケータを取得します

構文

```
'Declaration  
Public Read Property Locator As String
```

アクセス

読み取り専用。

TechDomains プロパティ

クラス

[BaseState](#)。

処理

実行可能ファイルに必要なテクノロジー ドメインを取得します。

構文

```
'Declaration  
Public Read Property TechDomains As List of TechDomain
```

アクセス

読み取り専用。

WorkingDirectory プロパティ

クラス

[BaseState](#)。

処理

作業ディレクトリを取得または設定します。Windows 環境変数を含めることができます。

構文

```
'Declaration  
Public ReadWrite Property WorkingDirectory As String
```

アクセス

読み書き。

BrowserBaseState クラス

説明

Executable で指定されたブラウザが適切に実行されていて、テストの準備ができていることを保証します。さらに、基本状態として Url によって指定された URL に移動し、ブラウザを前面に持ってきます。ブラウザの基本状態は、以下のように実行されます。

- エージェントをブラウザにアタッチします。
 - Locator で適切に指定されたテスト オブジェクトが見つかったら、基本状態として所定の URL に移動し、ブラウザが前面に配置され、見つかったテスト オブジェクトが返されます。
 - テスト オブジェクトが見つからない場合、browserBaseStateInfo で指定したとおりにブラウザがコマンドラインから起動し、エージェントがブラウザにアタッチされます。
- テスト オブジェクトが見つかるまで、エージェントは待機します。タイムアウトは、オプション Options.ObjectResolveTimeout。タイムアウトを変更するには、Desktop クラスの SetOption メソッドを参照してください。
 - 指定したタイムアウト内にテスト オブジェクトが見つかったら、基本状態として所定の URL に移動し、ブラウザが前面に配置され、見つかったテスト オブジェクトが返されます。
 - テスト オブジェクトが見つからずにタイムアウトした場合は、例外がスローされます。

継承

BrowserBaseState はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class BrowserBaseState _  
Implements IBaseState
```

プロパティ

名前	説明
CommandLineArguments	コマンドライン引数を取得または設定します。(BaseState から継承)。
CommandLinePattern	コマンドラインパターンを取得または設定します。(BaseState から継承)。
ExecutablePattern	実行可能パターンを取得または設定します。(BaseState から継承)。
Executable	実行可能ファイルの名前を取得します(BaseState から継承)。
Locator	実行可能ファイルのロケータを返します(BaseState から継承)。
TechDomains	テストに使用するテクノロジードメインのリストを取得または設定します(BaseState から継承)。
Url	移動する URL を取得します。
WorkingDirectory	作業ディレクトリを取得または設定します。(BaseState から継承)。

URL プロパティ

クラス

[BrowserBaseState](#)。

処理

移動する URL を取得します。

構文

```
'Declaration  
Public Read Property Url As String
```

アクセス

読み取り専用。

ConsoleWindow クラス

説明

ConsoleWindow はコンソールのクラスです。

継承

ConsoleWindow はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class ConsoleWindow
```

メソッド

名前	説明
Close	コンソールを閉じます。
CloseAll	すべての開いているコンソールまたは特定のキャプションを持つすべてのコンソールを閉じます。
GetContents	コンソールの内容を返します。
Open	新しいコンソールを開きます。
TypeKeys	一連のキーストロークをコンソールに送信します。

Close メソッド (ConsoleWindow)

クラス

[ConsoleWindow](#)。

処理

コンソールを閉じます。

構文

```
consoleWindow.Close ([sItemIdentifier])
```

変数	説明
<i>sItemIdentifier</i>	コンソール ウィンドウのインデックスまたはウィンドウのキャプション。このパラメータが指定されていない場合、Silk Test Workbench は最初に見つけたコンソール ウィンドウを閉じます。STRING。

例 1

次の例では、インデックスを使用して 2 番目のコンソール ウィンドウを閉じる方法を示します。

```
consoleWindow.Close(2)
```

例 2

次の例では、コンソール ウィンドウの完全識別子を使用してコンソール ウィンドウを閉じる方法を示します。

```
consoleWindow.Close("C:¥¥Windows¥¥system32¥¥cmd.exe[1]")
```

CloseAll メソッド (ConsoleWindow)

クラス

[ConsoleWindow](#)。

処理

すべての開いているコンソールまたは特定のキャプションを持つすべてのコンソールを閉じます。

構文

VB

```
consoleWindow.CloseAll ([sCaption])
```

変数	説明
<i>sCaption</i>	このパラメータを設定すると、このキャプションを持つコンソールだけが閉じられます。STRING

例

次の例では、キャプション MyCommandWindow を持つすべてのコンソール ウィンドウを閉じる方法を示します：

```
// VB  
consoleWindow.CloseAll("MyCommandWindow")
```

GetContents メソッド (ConsoleWindow)

クラス

[ConsoleWindow](#)。

処理

コンソールの内容を返します。

構文

```
IsContents = consoleWindow.GetContents ([sItemIdentifier])
```

変数	説明
<i>IsContents</i>	コンソールの内容。LIST OF STRING。
<i>sItemIdentifier</i>	コンソール ウィンドウのインデックスまたはウィンドウのキャプション。このパラメータが指定されていない場合、Silk Test Workbench は最初に見つけたコンソール ウィンドウの内容を返します。STRING。

メモ

- 返された文字列リスト内の各文字列は、コンソール内の 1 行に対応しています。

例 1

次の例では、インデックスを使用して 2 番目のコンソール ウィンドウの内容を取得する方法を示します。

```
IsContents = consoleWindow.GetContents(2)
```

例 2

次の例では、コンソール ウィンドウの完全識別子を使用してコンソール ウィンドウの内容を取得する方法を示します。

```
IsContents = consoleWindow.GetContents("C:¥¥Windows¥¥system32¥¥cmd.exe[1]")
```

Open メソッド (ConsoleWindow)

クラス

ConsoleWindow。

処理

新しいコンソールを開きます。

使用可能な資産

この機能がサポートされるのは、Open Agent を使用している場合のみです。

構文

VB

```
ConsoleWindow.Open ([caption])
```

変数	説明
caption	新しいコンソール ウィンドウのキャプション。String。

例

次の例では、キャプション "My Console" の新しいコンソールを開く方法について示します。

```
// VB  
ConsoleWindow.Open("My Console")
```

TypeKeys メソッド (ConsoleWindow)

クラス

[ConsoleWindow](#)。

処理

一連のキーストロークをコンソールに送信します。

構文

```
ConsoleWindow.TypeKeys (sEvents [, nDelay, ensureFocus, sItemIdentifier])
```

変数	説明
sEvents	入力するキーストローク。STRING。
nDelay	省略可能 : キーストローク間の遅延 (秒単位、小数可)。NUMBER。
ensureFocus	省略可能 : アプリケーションを自動的に前面に移動し、メソッドを実行したコントロールにキーボードのフォーカスを設定するかどうか。この処理は、最初のキーを入力する前に実行されるため、入力するキーを複数指定した場合、キーストロークごとに再確認されることはありません。この変数は、デフォルトで TRUE に設定されています。BOOLEAN。

変数	説明
<i>sItemIdentifier</i>	コンソール ウィンドウのインデックスまたはウィンドウのキャプション。このパラメータが指定されていない場合、Silk Test Workbench は最初に見つけたコンソール ウィンドウにキーストロークを送信します。STRING。

メモ

- 開いているコンソールがない場合は、TypeKeys によってコンソールが開きます。
- Windows の場合は、入力する前に TypeKeys によってコンソールに入力フォーカスが設定されます。
- *nDelay* を指定した場合、この値は、SetOption で設定された、OPT_KEYBOARD_DELAY オプションを使用した設定値よりも優先します。
- キーストロークが認識されない場合は、使用する遅延の値を大きくすることができます。
- キーの後に、そのキーの入力頻度を定義するカウンターを配置できます。カウンターは遅延と組み合わせることができます。
- 遅延時間中は画面がロックされ、マウスおよびキーボードとの対話は不可能になります。
- DOM に拡張機能を使用して記録している場合、TypeKeys("<Tab>") はキャプチャされません。入力するオブジェクトはスクリプトで直接参照されているため、この手動の Tab を記録する必要はありません。必要に応じて、スクリプトに TypeKeys("<Tab>") を手動で入力できますが、この Tab は記録されません。
- **Ctrl-Shift** などのキー グループを指定した場合、プラスおよびマイナス記号はサポートされます。

例

次のサンプル コードは、コンソール ウィンドウで dir コマンドを実行し、ディレクトリのファイルとフォルダをリストします。

```
// VB
ConsoleWindow.TypeKeys ("dir<Enter>")
```

DllCall クラス

説明

DllCall クラスを使用すると、NTF ユーザーはエージェントまたは AUT のいずれかで、任意の DLL の公開された DLL 機能呼び出すことができます。

継承

DllCall はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration
Public Class Agent
```

メソッド

名前	説明
UnloadAllDllsFromAgent	エージェントからすべての DLL をアンロードします。
UnloadDllFromAgent	指定した DLL をエージェントからアンロードします。

UnloadAllDllsFromAgent メソッド

クラス

[DllCall](#)。

処理

エージェントからすべての DLL をアンロードします。

構文

```
Agent.UnloadAllDllsFromAgent ()
```

UnloadDllFromAgent メソッド

クラス

[DllCall](#)。

処理

指定した DLL をエージェントからアンロードします。

構文

```
bSuccessful = Agent.UnloadDll (sDllName)
```

変数	説明
<i>bSuccessful</i>	DLL が正常にアンロードされた場合は true、失敗した場合は false。BOOLEAN。
<i>sDllName</i>	DLL ファイルの名前。STRING。

メモ

UnloadDllFromAgent は、標準 DLL をアンロードする場合のみ使用できます。混在アセンブリをアンロードする場合は、使用できません。混在アセンブリをアンロードするには、UnloadAllDllsFromAgent を使用します。

IBaseState インターフェイス

説明

基本状態のインターフェイスです。基本状態は、アプリケーションをテスト可能な状態にします。必要に応じてアプリケーションを起動して、あるロケータが見つかるまで待機し、アプリケーションを前面に持ってきます。

構文

```
'Declaration  
Public Interface IBaseState
```


メソッド

名前	説明
Execute	デスクトップで指定したマシン上で基本状態を実行します。

Execute メソッド

クラス

[BaseState](#)。

処理

デスクトップで指定したマシン上で基本状態を実行します。

構文

```
TestObject = BaseState.Execute()
```

変数	説明
TestObject	ロケーターが待機するオブジェクト。

例：基本状態でのブラウザと Web アプリケーションの URL の指定

Web アプリケーションをテストするブラウザとして Internet Explorer を、Web アプリケーションの URL として `www.borland.com` を指定するには、次のコードを使用できます。

```
Dim baseState = New BrowserBaseState(BrowserType.InternetExplorer, "www.borland.com")
baseState.Execute()
```

さらに、基本状態を変数に格納するには：

```
Dim baseState = New BrowserBaseState(BrowserType.InternetExplorer, "www.borland.com")
Dim browserApplication As BrowserApplication = baseState.Execute()
```

モバイルデバイス上のモバイルブラウザ (iOS デバイス上の Safari など) を指定するには、次のコードを使用できます。

```
Dim baseState = New BrowserBaseState(BrowserType.Safari, "www.borland.com")
baseState.MobileDeviceName = "My iPhone"
baseState.Execute()
```

例：実行可能ファイルへのパスと AUT のメイン ウィンドウのロケーターの指定

AUT の実行可能ファイルへのパスとして `C:/windows/system32/notepad.exe` を、AUT のメイン ウィンドウのロケーターとして `//Window[@caption='*Notepad*']` を指定するには、次のコードを使用できます。

```
Dim baseState = New BaseState("C:/windows/system32/notepad.exe", "//Window[@caption='*Notepad*']")
baseState.Execute()
```

さらに、基本状態を変数に格納するには：

```
Dim baseState = New BaseState("C:/windows/system32/notepad.exe", "//  
Window[@caption='*Notepad*']")  
Dim mainWindow As Window = baseState.Execute()
```

Timer クラス

説明

Timer クラスを使用すると、経過時間を正確に計測できます。

特に、Timer クラスのメソッドとプロパティは、Silk Performer から呼び出されるテスト実行の計測に使用できるという利点があります。Silk Test Workbench と Silk Performer の統合についての詳細は、『[Silk Performer ヘルプ](#)』を参照してください。

継承

BaseState はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class Timer
```

メソッド

名前	説明
Pause	タイマーの計測を一時停止します。
Resume	タイマーの計測を再開します。
Start	タイマーの計測を開始または再開します。
StartNew	新しい Timer インスタンスを初期化し、経過時間を 0 に設定してから、経過時間の計測を開始します。
Stop	タイマーの計測を停止します。

プロパティ

名前	説明
Elapsed	タイマーの経過時間を取得します。
ElapsedMilliseconds	タイマーの経過時間をミリ秒で取得します。
IsRunning	タイマーが現在実行されているかどうかを取得します。
Name	Silk Performer とのやり取りに使用するタイマーの名前を取得または設定します。

Pause メソッド (Timer)

クラス

[Timer](#)。

処理

タイマーの計測を一時停止します。

構文

```
timer.Pause ()
```

Resume メソッド (Timer)

クラス

[Timer](#)。

処理

タイマーの計測を再開します。

構文

```
timer.Resume ()
```

Start メソッド (Timer)

クラス

[Timer](#)。

処理

タイマーが現在停止している場合、タイマーの計測を開始します。タイマーが現在一時停止している場合、タイマーの計測を再開します。

構文

```
timer.Start ()
```

StartNew メソッド (Timer)

クラス

[Timer](#)。

処理

新しい Timer インスタンスを初期化し、経過時間を 0 に設定してから、経過時間の計測を開始します。名前は Silk Performer とやり取りするために提供されます。

構文

```
timer=StartNew ([timerName])
```

変数	説明
<code>timer</code>	経過時間の測定を開始した新しい Timer インスタンス。 Timer。

変数	説明
<code>timerName</code>	省略可能：新しいタイマーの名前。名前は Silk Performer とやり取りするために提供されます。String.

Stop メソッド (Timer)

クラス

[Timer](#)。

処理

タイマーの計測を停止します。

構文

```
timer.Stop()
```

Elapsed プロパティ (Timer)

クラス

[Timer](#)。

処理

タイマーの経過時間を取得します。

構文

```
'Declaration
Public Read Property Elapsed As TimeSpan
```

アクセス

読み取り

ElapsedMilliseconds プロパティ (Timer)

クラス

[Timer](#)。

処理

タイマーの経過時間をミリ秒で取得します。

構文

```
'Declaration
Public Read Property ElapsedMilliseconds As Long
```

アクセス

読み取り

IsRunning プロパティ (Timer)

クラス

[Timer](#)。

処理

タイマーが現在実行されているかどうかを取得します。タイマーが実行している場合は True を返します。

構文

```
'Declaration  
Public Read Property IsRunning As Boolean
```

アクセス

読み取り

Name プロパティ (Timer)

クラス

[Timer](#)。

処理

Silk Performer とのやり取りに使用するタイマーの名前を取得または設定します。

構文

```
'Declaration  
Public ReadWrite Property Name As String
```

アクセス

読み書き。

Workbench クラス

説明

Workbench は、検証を追加したり、子スクリプトを実行するための機能を提供するクラスです。

継承

Workbench はどのクラスからも派生しません。また、このクラスから派生するクラスはありません。

構文

```
'Declaration  
Public Class Workbench
```

Workbench のメソッドを呼び出すには、以下の構文を使用します。

```
Workbench.method()
```

プロパティ

名前	説明
Username	現在の Silk Test Workbench ユーザーの名前を返します。

メソッド

名前	説明
LoadActiveData	ActiveData オブジェクトを読み込んで、スクリプトで使用します。
ResultComment Method	結果 ウィンドウの 詳細 タブに、コメントをステップとして追加します。
RunScript	名前によって識別される子スクリプトを、パラメータを渡して実行します。
Verify	予想した値と実際の値が等しいかどうかを検証します。コメントを指定することもできます。
VerifyAsset	検証資産を実行します。

LoadActiveData メソッド

クラス


[Workbench](#)。

処理

スクリプトで使用する ActiveData 資産を読み込みます。

構文

```
Workbench.LoadActiveData("activeDataName", [sheetName, startRow, endRow, randomCount, readOnly])
```

変数	説明
<code>activeDataName</code>	スクリプトで使用する ActiveData 資産の名前。STRING。  注: 環境内で複数のプロジェクトを使用する場合、Common プロジェクト内にある、名前が一意でない ActiveData 資産を挿入するには、プロジェクト修飾子を使用します。詳細については、例を参照してください。
<code>sheetName</code>	省略可能：使用するシートの名前。ここで指定したシートは、ActiveData 資産で指定したシートを上書きします。デフォルトでは、資産で指定したシートが使用されます。STRING
<code>startRow</code>	省略可能：最初に読み込む ActiveData 資産内の行のインデックスです。INTEGER。
<code>endRow</code>	省略可能：最後に読み込む ActiveData 資産内の行のインデックスです。INTEGER。
<code>randomCount</code>	省略可能：資産に関連付けられた ActiveData ファイルのレコードを ActiveData テストで使用するかどうか、およびその方法を決定します。値が 0 または False の場合は、ランダム モードがオフになります。値が -1 または True の場合は、値が StartRow から EndRow のすべての行が ActiveData テストでランダムな順番で使用されます。値が 1 から N (N はテスト データ資産の行

変数	説明
<i>readOnly</i>	<p>数) の場合は、 StartRow から EndRow の範囲でのランダムな行数として使用されます。 RandomCount パラメータの許容値とともに、さまざまな StartRow 値および EndRow 値を使用する場合の詳細については、「アクティブデータに使用するデータの指定」を参照してください。 INTEGER。</p> <p>省略可能 : ActiveData 資産を読み取り/書き込み可能で開くか、読み取り専用で開くかを決定します。読み取り/書き込み可能で ActiveData 資産を開くには、False に設定します。ActiveData 資産を読み取り専用として開くには、True に設定します。読み取り専用を設定すると、スクリプトの ActiveData テストの部分で ActiveData 資産に書き込みを行おうとした場合に実行時エラーが生成されます。デフォルトでは、ActiveData 資産は読み取り/書き込み可能で開かれます。 BOOLEAN。</p>

使用例

スクリプトに ActiveData 内のランダムな行を含めるには、次のように入力します。

```
Public Sub Main()
  AddRandomToPhoneBook( 3 )    'Reads 3 rows in a random order
End Sub

Public Sub AddRandomToPhoneBook( howMany As Integer )
  Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData", 1, -1,
  howMany )
  Dim row As ActiveDataRow

  With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")
    For Each row In data
      Dim FirstName As String = row.GetString("fname")
      Dim LastName As String = row.GetString("lname")
      Dim PhoneNumber As String = row.GetString("number")

      .DomTextField("@id='txtFirstName']").SetText(FirstName)
      .DomTextField("@id='txtLastName']").SetText(LastName)
      .DomTextField("@id='txtPhoneNumber']").SetText(PhoneNumber)

      .DomButton("@id='btnAdd']").Select()
    Next
  End With
End Sub
```

スクリプトに ActiveData ファイル内のデータの一部を含めるには、次のように入力します。

```
Public Sub Main()
  AddSomeToPhoneBook( 3, 5 )    'Reads rows 3, 4 and 5 only
End Sub

Public Sub AddSomeToPhoneBook( firstToAdd As Integer, lastToAdd As Integer )
  Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData", firstToAdd,
  lastToAdd )
  Dim row As ActiveDataRow

  With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")
    For Each row In data
      Dim FirstName As String = row.GetString("fname")
      Dim LastName As String = row.GetString("lname")
      Dim PhoneNumber As String = row.GetString("number")
```

```

.DomTextField("@id='txtFirstName").SetText(FirstName)
.DomTextField("@id='txtLastName").SetText(LastName)
.DomTextField("@id='txtPhoneNumber").SetText(PhoneNumber)

.DomButton("@id='btnAdd").Select()
Next
End With
End Sub

```

スクリプトに ActiveData ファイル内のすべてのデータを読み取り専用として含めるには、次のように入力します。

```

Public Sub Main()
  AddAllToPhoneBookReadOnly() 'Opens the data file as read only
End Sub

Public Sub AddAllToPhoneBookReadOnly()
  Dim data As ActiveData = Workbench.LoadActiveData( "PhoneBookData", 1, -1, 0,
True )
  Dim row As ActiveDataRow

  With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")
    For Each row In data
      Dim FirstName As String = row.GetString("fname")
      Dim LastName As String = row.GetString("lname")
      Dim PhoneNumber As String = row.GetString("number")

      .DomTextField("@id='txtFirstName").SetText(FirstName)
      .DomTextField("@id='txtLastName").SetText(LastName)
      .DomTextField("@id='txtPhoneNumber").SetText(PhoneNumber)

      .DomButton("@id='btnAdd").Select()
    Next
  End With
End Sub

```

環境内で複数のプロジェクトを使用する場合、Common プロジェクト内にある、名前が一意でない ActiveData ファイルを挿入するには、プロジェクト修飾子を使用します。たとえば、以下のように入力します。

```

Public Sub Main()
  AddAllToPhoneBookReadOnly() 'Opens the data file as read only
End Sub

Public Sub AddAllToPhoneBookReadOnly()
  Dim data As ActiveData = Workbench.LoadActiveData( "Common.PhoneBookData", 1,
-1, 0, True )
  Dim row As ActiveDataRow


  With _desktop.BrowserWindow("/BrowserApplication//BrowserWindow")
    For Each row In data
      Dim FirstName As String = row.GetString("fname")
      Dim LastName As String = row.GetString("lname")
      Dim PhoneNumber As String = row.GetString("number")

      .DomTextField("@id='txtFirstName").SetText(FirstName)
      .DomTextField("@id='txtLastName").SetText(LastName)
      .DomTextField("@id='txtPhoneNumber").SetText(PhoneNumber)

      .DomButton("@id='btnAdd").Select()
    Next
  End With
End Sub

```


Next
End With
End Sub

 **注:** 任意のプロジェクト内の資産から、Common プロジェクト内の資産にアクセスできます。ただし、その逆はできません。現在処理中のプロジェクト内の ActiveData ファイルを使用する場合、プロジェクト修飾子を指定する必要はありません。別のプロジェクト内の Common プロジェクトにある ActiveData ファイルを使用する場合は、「Common」プロジェクト修飾子を指定します。

RunScript メソッド

クラス


[Workbench](#)。


処理

名前によって識別される子スクリプトを、パラメータを渡して実行します。

構文

```
Workbench.RunScript("scriptname", [parameters])
```

変数	説明
<code>scriptname</code>	実行する子スクリプトの名前。STRING。  注: 任意のプロジェクト内の資産から、Common プロジェクト内の資産にアクセスできます。ただし、その逆はできません。現在処理中のプロジェクト内のスクリプトを使用する場合、プロジェクト修飾子を指定する必要はありません。別のプロジェクト内の Common プロジェクトにあるスクリプトを使用する場合は、「Common」プロジェクト修飾子を指定します。
パラメータ	省略可能：子スクリプトからメイン スクリプトに渡すパラメーター。STRING または OBJECT。

 **注:** Oracle データベースに格納されるスクリプト名では、大文字と小文字が区別されます。Oracle データベースから挿入するスクリプトの名前は、データベースでの名前と正確に一致する必要があります。一致していないと、Silk Test Workbench でスクリプトの読み込みに失敗します。

使用例

環境内で複数のプロジェクトを使用する場合、Common プロジェクト内にある、名前が一意でないスクリプトを挿入するには、プロジェクト修飾子を使用します。たとえば、以下のように入力します。

```
Workbench.RunScript("Common.AddAccount")
```

ここで、`AddAccount` は Common プロジェクトおよび現在使用中のプロジェクト内にあるスクリプトです。ユーザーは現在のプロジェクトでなく、Common プロジェクト内のスクリプトを使用することができます。

Verify メソッド

クラス

[Workbench](#)。

処理

予想した値と実際の値が等しいかどうかを検証します。コメントを指定することもできます。

構文

このメソッドには多くのオーバーロードがあります。

```
result = Workbench.Verify(condition[, VerifyFlags])
```

または

```
result = Workbench.Verify(condition[, comment, VerifyFlags])
```

または

```
result = Workbench.Verify(expected, actual[, VerifyFlags])
```

または

```
result = Workbench.Verify(expected, actual[, VerifyFlags, comment])
```

変数	説明
<i>result</i>	検証が成功したかどうか。BOOLEAN。
<i>condition</i>	満たす必要がある条件。BOOLEAN。
<i>expected</i>	スクリプトから返されると予測される値。OBJECT。
<i>actual</i>	スクリプトから実際に返される値。OBJECT。
<i>comment</i>	省略可能：追加するコメント。STRING。
<i>VerifyFlags</i>	省略可能：検証の失敗時にスクリーンショットをキャプチャするかどうか。この列挙には、次の値を指定できます。 <ul style="list-style-type: none">• None• ScreenShotOnFailure VerifyFlags。

使用例

期待値と実際の値を比較して、コメントを追加するには、以下のように入力します。

```
Workbench.Verify(expected As Object, actual As Object, comment As String)
```

たとえば、`Workbench.Verify("red", "green", "checking colors")` は検証に失敗し、メッセージ「checking colors - 実際：[green]; 予想： [red]」が表示されます。

期待値と実際の値を比較して、コメントを追加し、さらに検証が失敗した場合にスクリーンショットを結果ファイルに追加するには、以下のように入力します。

```
Workbench.Verify(expected As Object, actual As Object, comment As String, verifyFlags As VerifyFlags)
```

たとえば、`Workbench.Verify("red", "green", "checking colors", verifyFlags.ScreenShotOnFailure)` は検証に失敗し、メッセージ「checking colors - 実際：[green]; 予想： [red]」が表示され、スクリーンショットが結果ファイルに追加されます。

予想結果によって返された値を検証して、コメントを追加するには、以下のように入力します。

```
Workbench.Verify(condition As Boolean, comment As String)
```

たとえば、`Workbench.Verify(True, "Test Passed")` は検証に合格します。一方、`Workbench.Verify(False, "Test Failed")` は検証に失敗します。

予想結果によって返された値を検証して、コメントを追加し、さらに検証が失敗した場合にスクリーンショットを結果ファイルに追加するには、以下のように入力します。

```
Workbench.Verify(condition As Boolean, comment As String, verifyFlags As VerifyFlags)
```

たとえば、`Workbench.Verify(True, "Test Passed", verifyFlags.ScreenShotOnFailure)` は検証に合格し、スクリーンショットは追加されません。一方、`Workbench.Verify(False, "Test Failed", ScreenShotOnFailure)` は検証に失敗し、スクリーンショットが結果ファイルに追加されます。

`IEnumerable` オブジェクト（リストや配列など）について実際の値と期待値を比較するには、以下のように入力します。


```
Workbench.Verify(expectedEnumerable, actualEnumerable)
```


例：

```
Dim selectedItemsList = listBox.SelectedItems ' we assume that a list with the items  
"red" and "blue" is returned  
Dim expectedItemsList = New List(Of String)()  
expectedItemsList.Add("red")  
expectedItemsList.Add("blue")
```

```
Workbench.Verify(selectedItemsList, expectedItemsList) ' verification passes
```

```
Dim expectedItemsArray = New String() { "red", "blue" }  
Workbench.Verify(selectedItemsList, expectedItemsArray) ' verification passes
```

 **注:** 2つの `IEnumerable` オブジェクトは、要素の数が同じであり、それらの要素が等しく、かつ順序が同じである場合に、等しいとみなされます。

 **注:** 浮動小数点数を含む数学演算では、2つの数値が、ユーザーの視点では等しいにもかかわらず、内部表現のために完全には同一ではないとみなされる場合があります。このため、浮動小数点数（単精度、倍精度）は、差異が 0.00001 未満である場合は等しいとみなされます。この値は、状況によっては正しくない場合があります。その場合は、代わりに `Verify(result As Boolean)` を使用して2つの値を比較します。

VerifyAsset メソッド (Workbench)

クラス

[Workbench](#)。

処理

検証資産を実行します。検証する UI オブジェクトが見つからない場合は、`ObjectNotFoundException` をスローします。

構文

```
result = Verification.VerifyAsset(verificationAsset[, verifyFlags])
```

変数	説明
<code>result</code>	検証が成功したかどうか。BOOLEAN。
<code>verificationAsset</code>	実行する検証資産の名前。STRING。
<code>VerifyFlags</code>	省略可能：検証の失敗時にスクリーンショットをキャプチャするかどうか。この列挙には、次の値を指定できます。 <ul style="list-style-type: none">• None• ScreenShotOnFailure

変数	説明
	VerifyFlags。

使用例

イメージ資産 *myImageAsset* の検証を実行するには、次のように入力します。

```
Workbench.VerifyAsset("myImageAsset")
```

イメージ資産 *myImageAsset* の検証を実行して、検証の失敗時にスクリーンショットを結果ファイルに追加するには、次のように入力します。

```
Workbench.VerifyAsset("myImageAsset", verifyFlags.ScreenShotOnFailure)
```

ResultComment メソッド

クラス

[Workbench](#)。

処理

結果 ウィンドウの **詳細** タブに、コメントをステップとして追加します。結果コメントを使用して、テストに関する情報をテスト結果に追加できます。テストを再生すると、**結果** ウィンドウの **詳細** タブにコメントがステップとして表示されます。

構文

```
Workbench.ResultComment(comment)
```

変数	説明
<i>comment</i>	結果に追加されるコメント。STRING。

使用例

テストでいくつかの挿入された値のフォーマット検証が実行されなかったという結果を指摘する場合、次のような行をスクリプトに追加できます：

```
Workbench.ResultComment("Note: This test did not verify that the format of the inserted value was correct.")
```

Username プロパティ (Workbench)

クラス

[Workbench](#)。

処理

現在の Silk Test Workbench ユーザーの名前を返します。

構文

```
'Declaration  
Protected Read Property Username As String
```

アクセス

読み取り専用。

Data Types

This section lists the various types of data.



Note: For information on the .NET framework data types, for example Integer, Boolean, String, and so on, refer to the MSDN. For information about the data types that you can use in visual tests, see *Variables in Visual Tests*.

ClickType Enumeration

Description

Specifies how to click on a text during a replay of a TextClick.

Syntax

```
Public Enumeration ClickType
```

Members

Name	Description
Left	A single click with the left mouse button.
Right	A single click with the right mouse button.
Middle	A single click with the middle mouse button.
LeftDouble	A double click with the left mouse button.
Press	Clicking and holding the left mouse button.
Release	Releasing the left mouse button.

Color Class

Description

Describes a color in terms of red, green, blue, and alpha channels.

Inheritance Hierarchy

Color does not derive from any class, and no classes derive from it.

Syntax

```
Public Class Color
```

Fields

Name	Description
<i>BLACK</i>	Gets the color that has the ARGB value of #FF000000.
<i>WHITE</i>	Gets the color that has the ARGB value of #FFFFFFFF.

Name	Description
<i>GREEN</i>	Gets the color that has the ARGB value of #FF00FF00.
<i>RED</i>	Gets the color that has the ARGB value of #FFFF0000.
<i>BLUE</i>	Gets the color that has the ARGB value of #FF0000FF.
<i>YELLOW</i>	Gets the color that has the ARGB value of #FFFFFF00.
<i>MAGENTA</i>	Gets the color that has the ARGB value of #FFFF00FF.
<i>CYAN</i>	Gets the color that has the ARGB value of #FF00FFFF.

Properties

Name	Description
<i>R</i>	Gets the value of the red ARGB channel of the color. Integer
<i>G</i>	Gets the value of the green ARGB channel of the color. Integer
<i>B</i>	Gets the value of the blue ARGB channel of the color. Integer
<i>A</i>	Gets the value of the alpha ARGB channel of the color. Integer

ItemIdentifier Data Type

Description

Specifies an item in controls such as list boxes, tab controls, or combo boxes. Items can be identified by the following:

- Text (String).
- Index (Integer).
- A combination of text and index (String).

A string value specifies the name of the item. The string can contain wildcard characters. You can also specify the index of the item as a string, in the form "[n]", where n is the numerical index of the item.

Example

The following examples show how you can select an item in a ListBox:

```
listbox.select(0) ' Selects the first item, which has the index 0.
listbox.select("[0]") ' Also selects the first item.
listbox.select("apple") ' Selects the first item with text "apple".
listbox.select("apple[1]") 'Selects the second item with the text "apple".
```

ItemPath Data Type

Description

The hierarchical path from the root of the tree to a specific item. You have to always specify the entire hierarchy. Each part in the hierarchy of the item path can either be a name, an index, or a combination of both. You can uniquely identify each item in the hierarchical path by specifying one or more of its attributes.

Items in a tree have the following attributes:

- The hierarchical relationship between the item and other items.
- The title or the caption of the item.
- The 0-based ordinal number of the item in relation to its siblings.

Example

The following examples show how you can identify unique items within a tree by path, index, or a combination of both path and index:

Specification	Item
"/Desktop/ Applications/ Games"	A unique item "Games" under a unique item "Applications" under a unique item "Desktop" at the outer level of the tree.
"/ Desktop/[2]/ [4]"	The fifth item under the third item under a unique item "Desktop" at the outer level of the tree.
"/Desktop/ Applications[2]]/Games"	A unique item "Games" under the third item with the text "Applications" under a unique item "Desktop" at the outer level of the tree.

ModifierKeys Enumeration

Description

Specifies the key modifiers that should be pressed during replay.

Syntax

Public Enumeration ModifierKeys

Members

Name	Description
None	No key modifiers.
Shift	The Shift modifier.
Alt	The Alt modifier.
Cmd	The Cmd modifier. Supported only for desktop browsers on macOS.
Control	The Ctrl modifier.

Name	Description
ControlAlt	The Ctrl+Alt modifiers.
ShiftControl	The Shift+Ctrl modifiers.
ShiftAlt	The Shift+Alt modifiers.
ShiftControlAlt	The Ctrl+Shift+Alt modifiers.

Example

To follow a link from a Microsoft Word document that you are currently editing, you have to press **Ctrl** and click on the link. You can do this as follows:

```
link.Click(MouseButton.Left, ClickPosition.Center, ModifierKeys.Control)
```



Note: All three parameters are required.

MouseButton Enumeration

Description

Defines values that specify the buttons on a mouse device.

Syntax

```
Public Enumeration MouseButton
```

Members

Name	Description
Left	The left mouse button. Left = 1.
Middle	The middle mouse button. Middle = 3.
Right	The right mouse button. Right = 2.

Point Class

Description

Specifies a point on the screen.

Inheritance Hierarchy

Point does not derive from any class, and no classes derive from it.

Syntax

```
Public Class Point
```

Properties

Name	Description
X	Gets the x coordinate.

Name	Description
<i>Y</i>	Gets the y coordinate.

Range Class

Description

Specifies a range with a minimum and a maximum.

Inheritance Hierarchy

Range does not derive from any class, and no classes derive from it.

Syntax

```
Public Class Range
```

Properties

Name	Description
<i>Minimum</i>	Gets the minimum.
<i>Maximum</i>	Gets the maximum.

Rectangle Class

Description

Specifies a rectangle on the screen.

Inheritance Hierarchy

Rectangle does not derive from any class, and no classes derive from it.

Syntax

```
Public Class Rectangle
```

Properties

Name	Description
<i>X</i>	Gets the x coordinate of the rectangle.
<i>Y</i>	Gets the y coordinate of the rectangle.
<i>Width</i>	Gets the width of the rectangle.
<i>Height</i>	Gets the height of the rectangle.
<i>Center</i>	Gets the center of the rectangle.
<i>IsEmpty</i>	Determines whether the rectangle is empty.

Methods

Name	Description
Contains(Rectangle)	Determines whether the current Rectangle contains the specified Rectangle. Boolean
Contains(Point)	Determines whether the current Rectangle contains the specified Point. Boolean
Contains(X,Y)	Determines whether the current Rectangle contains the specified coordinates. Boolean

TextPosition Class

Description

Specifies a position within a text field.

Inheritance Hierarchy

TextPosition does not derive from any class, and no classes derive from it.

Syntax

```
Public Class TextPosition
```

Properties

Name	Description
Line	Gets the line.
Column	Gets the column.

TextRange Class

Description

Specifies a range within a text field.

Inheritance Hierarchy

TextRange does not derive from any class, and no classes derive from it.

Syntax

```
Public Class TextRange
```

Properties

Name	Description
StartLine	Gets the start line of the range.
StartColumn	Gets the start column of the range.
EndLine	Gets the end line of the range.

Name	Description
EndColumn	Gets the end column of the range.

TreeContent Class

Description

Represents the content of a tree control. Every item in a tree is represented as a `TreeNode`.

Inheritance Hierarchy

`TreeContent` does not derive from any class, and no classes derive from it.

Syntax

```
Public Class TreeContent
```

Properties

Name	Description
RootNodes	Gets the root nodes of the corresponding tree.

TreeNode Class

Description

Represents a node in the `TreeContent`.

Inheritance Hierarchy

`TreeNode` does not derive from any class, and no classes derive from it.

Syntax

```
Public Class TreeNode
```

Properties

Name	Description
Children	Gets the child nodes of the <code>TreeNode</code> .
Text	Gets the text of the <code>TreeNode</code> .

Flex Class Reference

Lists the available classes for testing Flex controls. You can also customize Silk Test to test custom controls.

For additional information about the classes, see the *Apache Flex Developer's Guide*.



Note: Any known problems with Flex and Flex classes can be reproduced in Silk Test. For instance, dragging and dropping an element that uses the `FlexChart` class may not work if a chart element is clicked near its edges before you start dragging the element. This results in

a failed replay because the drag start coordinates can be incorrect. This is a known issue with dragging and dropping in Flex. For details about known issues with Flex, see the *Apache Flex Release Notes*.



Note: The FlexTitleWindow class is recorded and recognized by Silk Test as a FlexPanel (although it is defined as a separate class). This is a known issue in Flex. For details about known issues with Flex, see the *Apache Flex Release Notes*.

FlexAccordion Class

Description

An Accordion navigator container has a collection of child containers, but only one of them at a time is visible. It creates and manages navigator buttons (accordion headers), which you use to navigate between the children.

There is one navigator button associated with each child container, and each navigator button belongs to the Accordion container, not to the child. When the user clicks a navigator button, the associated child container is displayed. The transition to the new child uses an animation to make it clear to the user that one child is disappearing and a different one is appearing.

The Accordion container does not extend the ViewStack container, but it implements all the properties, methods, styles, and events of the ViewStack container, such as selectedIndex and selectedChild.

Inheritance Hierarchy

- [FlexContainer](#)
 - FlexAccordion

Syntax

```
'Declaration  
Public Class FlexAccordion _  
Inherits FlexContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)

Name	Description
<i>backgroundDisabledColor</i>	the background color of the component when it is disabled. The global default value is undefined. (Inherited from <i>FlexContainer</i>)
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class. (Inherited from <i>FlexContainer</i>)
<i>backgroundSize</i>	the percentage to change the image size to for the specified <i>backgroundImage</i> . (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>headerHeight</i>	the height of each accordion header, in pixels.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalGap</i>	the number of pixels between children in the horizontal direction. The default value is 8.

Name	Description
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>paddingBottom</i>	the number of pixels between the container's bottom border and its content area.
<i>paddingTop</i>	the number of pixels between the container's top border and its content area.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedChild</i>	a reference to the currently visible child container.
<i>selectedIndex</i>	the zero-based index of the currently visible child container.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
textSelectedColor	the color of selected text. The default value is 0x2B333C.
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalGap	the number of pixels between children in the vertical direction.
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when the selected child container changes.
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MouseScroll</i>	Dispatched when a mouse wheel is spun over an <i>InteractiveObject</i> instance. (Inherited from <i>FlexContainer</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Scroll</i>	Dispatched when the user manually scrolls the container. (Inherited from <i>FlexContainer</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified

Name	Description
	interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexAdvancedDataGrid Class

Description

Expands on the functionality of the standard DataGrid control to add data visualization features to your Apache Flex application. These features provide greater control of data display, data aggregation, and data formatting.

The AdvancedDataGrid control is like a List control except that it can show more than one column of data, making it suited for showing objects with multiple properties.

Inheritance Hierarchy

- [FlexListBase](#)
 - FlexAdvancedDataGrid
 - [FlexOLAPDataGrid](#)

Syntax

```
'Declaration
Public Class FlexAdvancedDataGrid _
Inherits FlexListBase
```

Properties

Name	Description
allowDragSelection	whether drag-selection is enabled. Drag-Selection is the ability to select an item by dragging into it as opposed to normal selection where you can't have the mouse button down when you mouse over the item you want to select.
allowMultipleSelection	whether you can allow more than one item to be selected at the same time. (Inherited from FlexListBase)
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
alternatingItemColors	the colors to use for the backgrounds of the items in the list. (Inherited from FlexListBase)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)

Name	Description
<i>backgroundDisabledColor</i>	the color of text in the component if it is disabled. The default value is 0xAAB3B3. (Inherited from <i>FlexListBase</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnCount</i>	the number of columns to be displayed in a <i>TileList</i> control or items in a <i>HorizontalList</i> control. (Inherited from <i>FlexListBase</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexListBase</i>)
<i>columnWidth</i>	the width of the control's columns. (Inherited from <i>FlexListBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dataTipField</i>	the name of the field in the data provider items to display as the data tip. (Inherited from <i>FlexListBase</i>)
<i>depthColors</i>	the array of colors used for the rows of each level of the navigation tree of the <i>AdvancedDataGrid</i> control, in descending order. The default value is undefined.
<i>disabledColor</i>	the color of text in the component if it is disabled.
<i>displayItemsExpanded</i>	whether the navigation tree is expanded to show all items.
<i>editable</i>	whether the user can edit items in the data provider.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexListBase</i>)
<i>focusAlpha</i>	the alpha transparency value of the focus skin. The default value is 0.4. (Inherited from <i>FlexListBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>headerColors</i>	an array of two colors used to draw the header background gradient.
<i>headerHeight</i>	the height of the header cell of the column, in pixels.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalGridLineColor</i>	the color of the horizontal grid lines.
<i>iconField</i>	the name of the field in the data provider object that determines what to display as the icon. (Inherited from <i>FlexListBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)

Name	Description
<i>indentation</i>	the indentation for each node of the navigation tree, in pixels. The default value is 17.
<i>labelField</i>	the name of the field in the data provider items to display as the label. (Inherited from <i>FlexListBase</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexListBase</i>)
<i>lockedColumnCount</i>	the index of the first column in the control that scrolls. (Inherited from <i>FlexListBase</i>)
<i>lockedRowCount</i>	the index of the first row in the control that scrolls. (Inherited from <i>FlexListBase</i>)
<i>minColumnWidth</i>	the minimum width of the columns, in pixels.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexListBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexListBase</i>)
<i>openDuration</i>	the length of an open or close transition for the navigation tree, in milliseconds. The default value is 250.
<i>paddingBottom</i>	the number of pixels between the bottom of the row and the bottom of the renderer in the row.
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2.
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0.
<i>paddingTop</i>	the number of pixels between the top of the row and the top of the renderer in the row.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizableColumns</i>	whether the user can change the size of the columns.

Name	Description
<i>rollOverColor</i>	the color of the background of a renderer when the user rolls over it. The default value is 0xEEFEE6. (Inherited from <i>FlexListBase</i>)
<i>rowCount</i>	the number of rows to be displayed. (Inherited from <i>FlexListBase</i>)
<i>rowHeight</i>	the height of the rows in pixels. (Inherited from <i>FlexListBase</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectable</i>	whether the list shows selected items as selected. (Inherited from <i>FlexListBase</i>)
<i>selectedCells</i>	an Array of cell locations as row and column indices.
<i>selectedIndex</i>	the index in the data provider of the selected item. (Inherited from <i>FlexListBase</i>)
<i>selectedIndices</i>	an array of indices in the data provider of the selected items. (Inherited from <i>FlexListBase</i>)
<i>selectedItem</i>	a reference to the selected item in the data provider. (Inherited from <i>FlexListBase</i>)
<i>selectedItems</i>	an array of references to the selected items in the data provider. (Inherited from <i>FlexListBase</i>)
<i>selectionColor</i>	the color of the background of a renderer when the user selects it. The default value is 0x7FCEFF. (Inherited from <i>FlexListBase</i>)
<i>selectionDisabledColor</i>	the color of the background of a renderer when the component is disabled. The default value is 0xDDDDDD. (Inherited from <i>FlexListBase</i>)
<i>selectionMode</i>	the selection mode of the control.
<i>showDataTips</i>	whether dataTips are displayed for text in the rows. (Inherited from <i>FlexListBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textRollOverColor</i>	the text color of the label as the user moves the mouse pointer over the button. The default value is 0x2B333C. (Inherited from <i>FlexListBase</i>)
<i>textSelectedColor</i>	the color of the text of a renderer when the user selects it. The default value is 0x2B333C. (Inherited from <i>FlexListBase</i>)

Name	Description
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
useRollOver	whether items are highlighted as the mouse rolls over them. (Inherited from FlexListBase)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
variableRowHeight	whether the individual rows can have different height. (Inherited from FlexListBase)
verticalAlign	the vertical alignment of a renderer in a row. (Inherited from FlexListBase)
verticalGridLineColor	the color of the vertical grid lines. The default value is 0x666666.
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
wordWrap	whether text in the row should be word wrapped. (Inherited from FlexListBase)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes or collapses a AdvancedDataGrid branch.
ColumnGroupedADGHeaderShift	Dispatched when the user releases the mouse button on a column header after having dragged the column to a new location resulting in shifting the column to a new index.
ColumnStretch	Dispatched when a user changes the width of a column, indicating that the amount of data displayed in that column may have changed.
Deselect	Defines the value of the type property of the event object for an event that is dispatched when a previously selected item is deselected. (Inherited from FlexListBase)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Defines the value of the type property of the ListEvent object for an itemDoubleClick event, which indicates that the user double-clicked the mouse over a visual item in the control. (Inherited from FlexListBase)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexListBase)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexListBase)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexListBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCellData	Returns data for a cell in the grid.
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetGroupedItemChildrenCount	Returns the number of children within the first item of a group.

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetRowData	Returns data for the row of the item in the data provider.
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexListBase)
HeaderClick	Dispatched when the user releases the mouse button on a column header to request the control to sort the grid contents based on the contents of the column.
HeaderShift	Dispatched when the user releases the mouse button on a column header after having dragged the column to a new location resulting in shifting the column to a new index.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsGroupedItem	Returns a value that specifies whether the item is a member of a group.
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexScrollBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
MultiSelect	Defines the value of the type property of the event object for an event that is dispatched when an item is selected as part of an action that selects multiple items. (Inherited from FlexListBase)
Open	Opens or expands an AdvancedDataGrid branch.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Defines the value of the type property of the event object for a scroll event. (Inherited from FlexListBase)
ScrollToIndex	Ensures that the data provider item at the given index is visible. (Inherited from FlexListBase)
Select	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected. (Inherited from FlexListBase)
SelectIndex	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected. (Inherited from FlexListBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexListBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexAlert Class

Description

A pop-up dialog box that can contain a message, a title, buttons (any combination of OK, Cancel, Yes, and No) and an icon.

The Alert control is modal, which means it will retain focus until the user closes it.

Inheritance Hierarchy

- [FlexPanel](#)
 - FlexAlert

Syntax

```
'Declaration
Public Class FlexAlert _
Inherits FlexPanel
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)
backgroundDisabledColor	the background color of the component when it is disabled. The global default value is undefined. (Inherited from FlexContainer)
backgroundImage	the background image of a component. This can be an absolute or relative URL or class. (Inherited from FlexContainer)

Name	Description
<i>backgroundSize</i>	the percentage to change the image size to for the specified backgroundImage. (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>defaultButtonFlag</i>	A bitflag that contains either Alert.OK, Alert.CANCEL, Alert.YES, or Alert.NO to specify the default button. The default value is Alert.OK.
<i>dropShadowEnabled</i>	whether the Panel container's drop shadow is visible. The default value is true. (Inherited from <i>FlexPanel</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>footerColors</i>	the array of two colors used to draw the footer (area for the ControlBar container) background. (Inherited from <i>FlexPanel</i>)
<i>headerColors</i>	the array of two colors used to draw the header. (Inherited from <i>FlexPanel</i>)

Name	Description
<i>headerHeight</i>	the height of the header. The default value is based on the style of the title text. (Inherited from <i>FlexPanel</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>status</i>	the text in the status area of the title bar. The default value is "". (Inherited from <i>FlexPanel</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
text	The text to display in this alert dialog box. The default value is "".
themeColor	the theme color of a component. (Inherited from FlexObject)
title	the title or caption displayed in the title bar. The default value is "". (Inherited from FlexPanel)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MouseScroll</i>	Dispatched when a mouse wheel is spun over an <i>InteractiveObject</i> instance. (Inherited from <i>FlexContainer</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Scroll</i>	Dispatched when the user manually scrolls the container. (Inherited from <i>FlexContainer</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified

Name	Description
	interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexApplication Class

Description

A default, or Application, container that lets you start adding content to your application without explicitly defining another container.

Inheritance Hierarchy

- [FlexContainer](#)
 - FlexApplication
 - [FlexWindowedApplication](#)

Syntax

```
'Declaration
Public Class FlexApplication _
Inherits FlexContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the <code>backgroundColor</code> property, of the image or SWF file defined by the <code>backgroundImage</code> style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)
backgroundDisabledColor	the background color of the component when it is disabled. The global default value is undefined. (Inherited from FlexContainer)
backgroundImage	the background image of a component. This can be an absolute or relative URL or class. (Inherited from FlexContainer)

Name	Description
<i>backgroundSize</i>	the percentage to change the image size to for the specified backgroundImage. (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If

Name	Description
	mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numChildren	the number of child components in this container. (Inherited from FlexContainer)
numColumns	the total number of columns of data available. (Inherited from FlexContainer)
numRows	the total number of rows of data available. (Inherited from FlexContainer)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
url	the URL from which this Application's SWF file was loaded.
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)

Name	Description
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexAreaChart Class

Description

Represents data as an area bounded by a line connecting the values in the data. The AreaChart control can be used to represent different variations, including simple areas, stacked, 100% stacked, and high/low.

Inheritance Hierarchy

- [FlexCartesianChart](#)
 - FlexAreaChart

Syntax

```
'Declaration
Public Class FlexAreaChart _
Inherits FlexCartesianChart
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>clipContent</i>	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered. (Inherited from <i>FlexChart</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexChart</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dataTipMode</i>	how Flex displays DataTip controls for the chart. (Inherited from <i>FlexChart</i>)
<i>description</i>	a short description of the data in the chart. (Inherited from <i>FlexChart</i>)
<i>dragEnabled</i>	whether you can drag items out of this chart and drop them on other controls. (Inherited from <i>FlexChart</i>)
<i>dropEnabled</i>	whether dragged items can be dropped onto the chart. (Inherited from <i>FlexChart</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChart</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexChart</i>)
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11. (Inherited from <i>FlexChart</i>)

Name	Description
<i>gutterBottom</i>	The size of the region, in pixels, between the bottom of the chart data area and the bottom of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterLeft</i>	The size of the region, in pixels, between the left of the chart data area and the left of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterRight</i>	The size of the region, in pixels, between the right side of the chart data area and the outside of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterTop</i>	The size of the region, in pixels, between the top of the chart data area and the top of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAxisRatio</i>	the height limit of the horizontal axis. (Inherited from <i>FlexCartesianChart</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChart</i>)
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show. (Inherited from <i>FlexChart</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>mouseSensitivity</i>	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart. (Inherited from <i>FlexChart</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChart</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChart</i>)
<i>paddingBottom</i>	the number of pixels between the chart's bottom border and its content area. (Inherited from <i>FlexChart</i>)
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from <i>FlexChart</i>)

Name	Description
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from <i>FlexChart</i>)
<i>paddingTop</i>	the number of pixels between the chart's top border and its content area. (Inherited from <i>FlexChart</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedChartItem</i>	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item. (Inherited from <i>FlexChart</i>)
<i>selectedChartItems</i>	an array of all the selected ChartItem objects in the chart. (Inherited from <i>FlexChart</i>)
<i>selectionMode</i>	whether ChartItem objects can be selected. (Inherited from <i>FlexChart</i>)
<i>showDataTips</i>	whether DataTip controls for the chart show. (Inherited from <i>FlexChart</i>)
<i>showDataTipTargets</i>	whether to show targets over the datapoints when showDataTips is set to true. (Inherited from <i>FlexChart</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TextObject</i>)
<i>textAlign</i>	the alignment of text within a container. (Inherited from <i>FlexCartesianChart</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAxisRatio	the width limit of the vertical axis. (Inherited from FlexCartesianChart)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexChart)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexChart)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexChart)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChart)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header. (Inherited from FlexChart)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
WaitForObject	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexAreaSeries Class

Description

Defines a data series for an `AreaChart` control.

Inheritance Hierarchy

- [FlexChartSeries](#)
 - `FlexAreaSeries`

Syntax

```
'Declaration
Public Class FlexAreaSeries _
Inherits FlexChartSeries
```


Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexChartSeries</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChartSeries</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>form</i>	the boundary type for the area.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChartSeries</i>)
<i>minField</i>	the field of the dataProvider that determines the bottom boundary of the area.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChartSeries</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChartSeries</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>radius</i>	the radius, in pixels, of the chart elements for the data points.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index of the selected item in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedIndices</i>	an array of indexes of the selected items in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedItem</i>	the chart item that is selected in the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedItems</i>	an array of chart items that are selected in the series. (Inherited from <i>FlexChartSeries</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)

Name	Description
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
xField	the field of the data provider that determines the position of the data points on the horizontal axis.
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
yField	the field of the data provider that determines the position of the data point on the vertical axis.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChartSeries)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
WaitForObject	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexAxisRenderer Class

Description

Describe the horizontal and vertical axes of a chart. An axis is responsible for rendering the labels, tick marks, and title along its length.

AxisRenderer objects inherit some of their visual properties from the enclosing chart object. The text format of the labels and title defaults to the CSS text properties of the renderer. You can control the formatting of the axis title separately by specifying a `axisTitleStyleName`, either on the AxisRenderer or on the enclosing chart.

Inheritance Hierarchy

- [FlexObject](#)

- FlexAxisRenderer

Syntax

'Declaration

Public Class FlexAxisRenderer _

Inherits FlexObject

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>canDropLabels</i>	whether the AxisRenderer should drop labels as necessary to lay out correctly.
<i>canStagger</i>	whether to stagger labels on two label rows. Use this setting to minimize the space required for the labels. The default value is true, which staggers the labels.
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>heightLimit</i>	the maximum amount of space, in pixels, that an axis renderer will take from a chart.
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>labelGap</i>	the gap between the end of the tick marks and the top of the labels, in pixels. The default value is 3.

Name	Description
<i>labelRotation</i>	the label rotation.
<i>lastVisibleRow</i>	the index of the last visible child.
<i>length</i>	the length of the axis, in screen coordinates.
<i>minorTickLength</i>	the length of the minor tick marks on the axis, in pixels. The default value is 0.
<i>minorTickPlacement</i>	a value where to draw the minor tick marks. Values include: <i>inside</i> , <i>outside</i> , <i>cross</i> , and <i>none</i> .
<i>minorTicks</i>	an array that specifies where Flex draws the minor tick marks along the axis. Each array element contains a value between 0 and 1.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available.
<i>numRows</i>	the total number of rows of data available.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>placement</i>	the side of the chart that the axisRenderer appears on.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>showLabels</i>	whether labels appear along the axis. The default value is true.
<i>showLine</i>	whether to display the axis. The default value is true.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
tickLength	the length of the tick marks on the axis, in pixels. The default value is 3.
tickPlacement	a value where to draw the tick marks. Values include: <i>inside</i> , <i>outside</i> , <i>cross</i> , and <i>none</i> .
ticks	an array that specifies where Flex draws the tick marks along the axis. Each array element contains a value between 0 and 1.
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexBarChart Class

Description

Represents data as a series of horizontal bars whose length is determined by values in the data. A `BarChart` control can represent different chart variations, including simple bars, clustered bars, stacked, 100% stacked, and high/low.

Inheritance Hierarchy

- [FlexCartesianChart](#)
 - FlexBarChart

Syntax

```
'Declaration  
Public Class FlexBarChart _  
Inherits FlexCartesianChart
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
barWidthRatio	how wide to draw the bars relative to the category width, as a percentage in the range of 0 to 1.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipContent	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered. (Inherited from FlexChart)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChart)
currentState	the current view state of the component. (Inherited from FlexObject)
dataTipMode	how Flex displays DataTip controls for the chart. (Inherited from FlexChart)
description	a short description of the data in the chart. (Inherited from FlexChart)
dragEnabled	whether you can drag items out of this chart and drop them on other controls. (Inherited from FlexChart)
dropEnabled	whether dragged items can be dropped onto the chart. (Inherited from FlexChart)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)

Name	Description
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChart</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexChart</i>)
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11. (Inherited from <i>FlexChart</i>)
<i>gutterBottom</i>	The size of the region, in pixels, between the bottom of the chart data area and the bottom of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterLeft</i>	The size of the region, in pixels, between the left of the chart data area and the left of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterRight</i>	The size of the region, in pixels, between the right side of the chart data area and the outside of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterTop</i>	The size of the region, in pixels, between the top of the chart data area and the top of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAxisRatio</i>	the height limit of the horizontal axis. (Inherited from <i>FlexCartesianChart</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChart</i>)
<i>maxBarWidth</i>	how wide to draw the bars, in pixels.
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show. (Inherited from <i>FlexChart</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>mouseSensitivity</i>	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart. (Inherited from <i>FlexChart</i>)

Name	Description
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChart</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChart</i>)
<i>paddingBottom</i>	the number of pixels between the chart's bottom border and its content area. (Inherited from <i>FlexChart</i>)
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from <i>FlexChart</i>)
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from <i>FlexChart</i>)
<i>paddingTop</i>	the number of pixels between the chart's top border and its content area. (Inherited from <i>FlexChart</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedChartItem</i>	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item. (Inherited from <i>FlexChart</i>)
<i>selectedChartItems</i>	an array of all the selected ChartItem objects in the chart. (Inherited from <i>FlexChart</i>)
<i>selectionMode</i>	whether ChartItem objects can be selected. (Inherited from <i>FlexChart</i>)
<i>showDataTips</i>	whether DataTip controls for the chart show. (Inherited from <i>FlexChart</i>)
<i>showDataTipTargets</i>	whether to show targets over the datapoints when showDataTips is set to true. (Inherited from <i>FlexChart</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
textAlign	the alignment of text within a container. (Inherited from FlexCartesianChart)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAxisRatio	the width limit of the vertical axis. (Inherited from FlexCartesianChart)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexChart)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexChart)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexChart)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChart)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header. (Inherited from FlexChart)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexBarSeries Class

Description

Defines a data series for a BarChart control.

Inheritance Hierarchy

- [FlexChartSeries](#)
 - FlexBarSeries

Syntax

```
'Declaration
Public Class FlexBarSeries _
Inherits FlexChartSeries
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
barWidthRatio	the width to render the bars relative to the category width.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChartSeries)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)

Name	Description
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChartSeries</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChartSeries</i>)
<i>maxBarWidth</i>	the width of the bars, in pixels.
<i>minField</i>	the field of the data provider that determines the bottom of each bar.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChartSeries</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChartSeries</i>)
<i>offset</i>	the amount to offset the center of the bars from the center of the available space, relative to the category width.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index of the selected item in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)

Name	Description
selectedIndices	an array of indexes of the selected items in the data provider of the series. (Inherited from FlexChartSeries)
selectedItem	the chart item that is selected in the series. (Inherited from FlexChartSeries)
selectedItems	an array of chart items that are selected in the series. (Inherited from FlexChartSeries)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TextObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
xField	the field of the data provider that determines the x-axis location of the top of each bar.
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
yField	the field of the data provider that determines the y-axis location of the bottom of each bar in the chart.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChartSeries)
HighlightObject	Highlights this object. (Inherited from TestObject)

Name	Description
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexBox Class

Description

A Box container lays out its children in a single vertical column or a single horizontal row.

Inheritance Hierarchy

- [FlexContainer](#)
 - FlexBox
 - [FlexDividedBox](#)
 - [FlexNavigationBar](#)

Syntax

```
'Declaration
Public Class FlexBox _
Inherits FlexContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)

Name	Description
<i>backgroundDisabledColor</i>	the background color of the component when it is disabled. The global default value is undefined. (Inherited from <i>FlexContainer</i>)
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class. (Inherited from <i>FlexContainer</i>)
<i>backgroundSize</i>	the percentage to change the image size to for the specified <i>backgroundImage</i> . (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>direction</i>	whether to use vertical (default) or horizontal layout.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAlign</i>	the horizontal alignment of children in the container.

Name	Description
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)

Name	Description
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAlign	the vertical alignment of children in the container.
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MouseScroll</i>	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from <i>FlexContainer</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Scroll</i>	Dispatched when the user manually scrolls the container. (Inherited from <i>FlexContainer</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>ToString</i>	Returns a string representation of the control. (Inherited from <i>FlexDisplayObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexBubbleChart Class

Description

Represents data with three values for each data point. Each data point is defined by a value determining its position along the horizontal axis, a value determining its position along the vertical axis, and a value determining the size of the chart element, relative to the other data points on the chart.

Inheritance Hierarchy

- [FlexCartesianChart](#)
 - FlexBubbleChart

Syntax

```
'Declaration  
Public Class FlexBubbleChart _  
Inherits FlexCartesianChart
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipContent	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered. (Inherited from FlexChart)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChart)
currentState	the current view state of the component. (Inherited from FlexObject)
dataTipMode	how Flex displays DataTip controls for the chart. (Inherited from FlexChart)
description	a short description of the data in the chart. (Inherited from FlexChart)
dragEnabled	whether you can drag items out of this chart and drop them on other controls. (Inherited from FlexChart)
dropEnabled	whether dragged items can be dropped onto the chart. (Inherited from FlexChart)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)

Name	Description
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChart</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexChart</i>)
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11. (Inherited from <i>FlexChart</i>)
<i>gutterBottom</i>	The size of the region, in pixels, between the bottom of the chart data area and the bottom of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterLeft</i>	The size of the region, in pixels, between the left of the chart data area and the left of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterRight</i>	The size of the region, in pixels, between the right side of the chart data area and the outside of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterTop</i>	The size of the region, in pixels, between the top of the chart data area and the top of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAxisRatio</i>	the height limit of the horizontal axis. (Inherited from <i>FlexCartesianChart</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChart</i>)
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show. (Inherited from <i>FlexChart</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>mouseSensitivity</i>	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart. (Inherited from <i>FlexChart</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChart</i>)

Name	Description
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChart</i>)
<i>paddingBottom</i>	the number of pixels between the chart's bottom border and its content area. (Inherited from <i>FlexChart</i>)
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from <i>FlexChart</i>)
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from <i>FlexChart</i>)
<i>paddingTop</i>	the number of pixels between the chart's top border and its content area. (Inherited from <i>FlexChart</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedChartItem</i>	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item. (Inherited from <i>FlexChart</i>)
<i>selectedChartItems</i>	an array of all the selected ChartItem objects in the chart. (Inherited from <i>FlexChart</i>)
<i>selectionMode</i>	whether ChartItem objects can be selected. (Inherited from <i>FlexChart</i>)
<i>showDataTips</i>	whether DataTip controls for the chart show. (Inherited from <i>FlexChart</i>)
<i>showDataTipTargets</i>	whether to show targets over the datapoints when showDataTips is set to true. (Inherited from <i>FlexChart</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TextObject</i>)
<i>textAlign</i>	the alignment of text within a container. (Inherited from <i>FlexCartesianChart</i>)

Name	Description
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAxisRatio	the width limit of the vertical axis. (Inherited from FlexCartesianChart)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexChart)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexChart)

Name	Description
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexChart)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChart)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header. (Inherited from FlexChart)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexBubbleSeries Class

Description

Defines a data series for a BubbleChart control.

Inheritance Hierarchy

- [FlexChartSeries](#)
 - FlexBubbleSeries

Syntax

```
'Declaration  
Public Class FlexBubbleSeries _  
Inherits FlexChartSeries
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChartSeries)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from FlexChartSeries)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)

Name	Description
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from FlexChartSeries)
<i>maxRadius</i>	The radius of the largest item rendered in this series.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from FlexChartSeries)
<i>numRows</i>	the total number of rows of data available. (Inherited from FlexChartSeries)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
<i>radiusField</i>	the field of the data provider that determines the radius of each symbol, relative to the other data points in the chart.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
<i>selectedIndex</i>	the index of the selected item in the data provider of the series. (Inherited from FlexChartSeries)
<i>selectedIndices</i>	an array of indexes of the selected items in the data provider of the series. (Inherited from FlexChartSeries)
<i>selectedItem</i>	the chart item that is selected in the series. (Inherited from FlexChartSeries)
<i>selectedItems</i>	an array of chart items that are selected in the series. (Inherited from FlexChartSeries)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)

Name	Description
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
xField	the field of the data provider that determines the x-axis location of each data point.
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
yField	the field of the data provider that determines the y-axis location of each data point.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChartSeries)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexButton Class

Description

The Button control is a commonly used rectangular button. Button controls look like they can be pressed. They can have a text label, an icon, or both on their face.

Buttons typically use event listeners to perform an action when the user selects the control. When a user clicks the mouse on a Button control, and the Button control is enabled, it dispatches a click event and a buttonDown event. A button always dispatches events such as the mouseMove, mouseOver, mouseOut, rollOver, rollOut, mouseDown, and mouseUp events whether enabled or disabled.

You can customize the look of a Button control and change its functionality from a push button to a toggle button. You can change the button appearance by using a skin for each of the button's states.

The label of a Button control uses a bold typeface. If you embed a font that you want to use for the label of the Button control, you must embed the bold typeface.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexButton
 - [FlexCheckBox](#)
 - [FlexPopUpButton](#)
 - [FlexRadioButton](#)

Syntax

```
'Declaration
Public Class FlexButton _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)

Name	Description
<i>color</i>	the color of text in the component, including the component label. The default value is 0x0B333C.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>disabledColor</i>	the color of text in the component if it is disabled. The default value is 0xAAB3B3.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alphas used for the background fill of controls.
<i>fillColors</i>	the colors used to tint the background of the control.
<i>focusAlpha</i>	the alpha transparency value of the focus skin. The default value is 0.4.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use.
<i>fontSize</i>	the height of the text, in pixels.
<i>fontStyle</i>	whether the text is italic font.
<i>fontWeight</i>	whether the text is boldface.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text to appear on the Button control.
<i>labelPlacement</i>	the orientation of the label in relation to a specified icon.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)

Name	Description
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selected</i>	whether a toggle button is toggled on (true) or off (false).
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textAlign</i>	the alignment of text within a container.
<i>textDecoration</i>	whether the text is underlined.
<i>textIndent</i>	the offset of the first line of text from the left side of the container, in pixels. The default value is 0.
<i>textRollOverColor</i>	the text color of the label as the user moves the mouse pointer over the button. The default value is 0x2B333C.
<i>textSelectedColor</i>	the text color of the label as the user presses it. The default value is 0x000000.
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toggle</i>	whether a Button is in a toggle state or not.
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter,

Name	Description
WaitForScreenshotStable	<p>the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexButtonBar Class

Description

Defines a horizontal or vertical group of logically related push buttons with a common look and navigation.

Inheritance Hierarchy

- [FlexNavigationBar](#)
 - FlexButtonBar
 - [FlexToggleButtonBar](#)

Syntax

```
'Declaration
Public Class FlexButtonBar _
Inherits FlexNavigationBar
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)

Name	Description
<i>backgroundDisabledColor</i>	the background color of the component when it is disabled. The global default value is undefined. (Inherited from <i>FlexContainer</i>)
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class. (Inherited from <i>FlexContainer</i>)
<i>backgroundSize</i>	the percentage to change the image size to for the specified <i>backgroundImage</i> . (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>direction</i>	whether to use vertical (default) or horizontal layout. (Inherited from <i>FlexBox</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>horizontalAlign</i>	the horizontal alignment of children in the container. (Inherited from <i>FlexBox</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index of the active navigation item, where the first item is at an index of 0. The default value is -1. (Inherited from <i>FlexNavigationBar</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAlign	the vertical alignment of children in the container. (Inherited from FlexBox)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Defines the value of the type property of the event object for an itemClick event. (Inherited from FlexNavigationBar)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)

Name	Description
<i>DragDrop</i>	Defines the value of the type property of the event object for a dragDrop event. (Inherited from <i>FlexContainer</i>)
<i>DragStart</i>	Defines the value of the type property of the event object for a dragStart event. (Inherited from <i>FlexContainer</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from <i>FlexDisplayObject</i>)
<i>GetValues</i>	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from <i>FlexContainer</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexNavigationBar)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexCandlestickChart Class

Description

Represents financial data as a series of candlesticks representing the high, low, opening, and closing values of a data series.

The top and bottom of the vertical line in each candlestick represent the high and low values for the datapoint, while the top and bottom of the filled box represents the opening and closing values. Each candlestick is filled differently depending on whether the closing value for the datapoint is higher or lower than the opening value.

Inheritance Hierarchy

- [FlexCartesianChart](#)
 - FlexCandlestickChart

Syntax

```
'Declaration
Public Class FlexCandlestickChart _
Inherits FlexCartesianChart
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipContent	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered. (Inherited from FlexChart)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChart)

Name	Description
<i>columnWidthRatio</i>	a ratio of how wide to draw the candlesticks relative to the horizontal axis's category widths, as a percentage in the range of 0 to 1.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dataTipMode</i>	how Flex displays DataTip controls for the chart. (Inherited from <i>FlexChart</i>)
<i>description</i>	a short description of the data in the chart. (Inherited from <i>FlexChart</i>)
<i>dragEnabled</i>	whether you can drag items out of this chart and drop them on other controls. (Inherited from <i>FlexChart</i>)
<i>dropEnabled</i>	whether dragged items can be dropped onto the chart. (Inherited from <i>FlexChart</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChart</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexChart</i>)
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11. (Inherited from <i>FlexChart</i>)
<i>gutterBottom</i>	The size of the region, in pixels, between the bottom of the chart data area and the bottom of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterLeft</i>	The size of the region, in pixels, between the left of the chart data area and the left of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterRight</i>	The size of the region, in pixels, between the right side of the chart data area and the outside of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterTop</i>	The size of the region, in pixels, between the top of the chart data area and the top of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAxisRatio</i>	the height limit of the horizontal axis. (Inherited from <i>FlexCartesianChart</i>)

Name	Description
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChart</i>)
<i>maxColumnWidth</i>	how wide to draw the candlesticks, in pixels.
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show. (Inherited from <i>FlexChart</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>mouseSensitivity</i>	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart. (Inherited from <i>FlexChart</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChart</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChart</i>)
<i>paddingBottom</i>	the number of pixels between the chart's bottom border and its content area. (Inherited from <i>FlexChart</i>)
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from <i>FlexChart</i>)
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from <i>FlexChart</i>)
<i>paddingTop</i>	the number of pixels between the chart's top border and its content area. (Inherited from <i>FlexChart</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)

Name	Description
<i>selectedChartItem</i>	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item. (Inherited from <i>FlexChart</i>)
<i>selectedChartItems</i>	an array of all the selected ChartItem objects in the chart. (Inherited from <i>FlexChart</i>)
<i>selectionMode</i>	whether ChartItem objects can be selected. (Inherited from <i>FlexChart</i>)
<i>showDataTips</i>	whether DataTip controls for the chart show. (Inherited from <i>FlexChart</i>)
<i>showDataTipTargets</i>	whether to show targets over the datapoints when showDataTips is set to true. (Inherited from <i>FlexChart</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TextObject</i>)
<i>textAlign</i>	the alignment of text within a container. (Inherited from <i>FlexCartesianChart</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TextObject</i>)
<i>verticalAxisRatio</i>	the width limit of the vertical axis. (Inherited from <i>FlexCartesianChart</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexChart)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexChart)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexChart)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChart)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header. (Inherited from FlexChart)

Name	Description
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by

Name	Description
WaitForPropertyNotEquals	the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexCandlestickSeries Class

Description

Represents financial data as a series of candlesticks representing the high, low, opening, and closing values of a data series.

The top and bottom of the vertical line in each candlestick represent the high and low values for the datapoint, while the top and bottom of the filled box represent the opening and closing values. Each candlestick is filled differently depending on whether the closing value for the datapoint is higher or lower than the opening value.

Inheritance Hierarchy

- [FlexHLOCSeriesBase](#)
 - FlexCandlestickSeries

Syntax

```
'Declaration
Public Class FlexCandlestickSeries _
Inherits FlexHLOCSeriesBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)

Name	Description
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexChartSeries</i>)
<i>columnWidthRatio</i>	the width of elements relative to the category width. (Inherited from <i>FlexHLOCSeriesBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChartSeries</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChartSeries</i>)
<i>maxColumnWidth</i>	the width of the elements, in pixels. (Inherited from <i>FlexHLOCSeriesBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChartSeries</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChartSeries</i>)

Name	Description
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index of the selected item in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedIndices</i>	an array of indexes of the selected items in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedItem</i>	the chart item that is selected in the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedItems</i>	an array of chart items that are selected in the series. (Inherited from <i>FlexChartSeries</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
xField	the field of the data provider that determines the x-axis location of the element. (Inherited from FlexHLOCSeriesBase)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChartSeries)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT

Name	Description
	takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexCanvas Class

Description

A Canvas layout container defines a rectangular region in which you place child containers and controls. It is the only container that lets you explicitly specify the location of its children within the container by using the x and y properties of each child.

Flex sets the children of a Canvas layout container to their preferred width and preferred height. You may override the value for a child's preferred width by setting its width property to either a fixed pixel value or a percentage of the container size. You can set the preferred height in a similar manner.

If you use percentage sizing inside a Canvas container, some of your components may overlap. If this is not the effect you want, plan your component positions and sizes carefully.

Inheritance Hierarchy

- [FlexContainer](#)
 - FlexCanvas

Syntax

```
'Declaration
Public Class FlexCanvas _
Inherits FlexContainer
```


Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)
backgroundDisabledColor	the background color of the component when it is disabled. The global default value is undefined. (Inherited from FlexContainer)
backgroundImage	the background image of a component. This can be an absolute or relative URL or class. (Inherited from FlexContainer)
backgroundSize	the percentage to change the image size to for the specified backgroundImage . (Inherited from FlexContainer)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexContainer)
creationIndex	the order to instantiate and draw the children of the container. (Inherited from FlexContainer)
creationPolicy	the child creation policy for this Container. (Inherited from FlexContainer)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
fillAlphas	the alpha transparency values used for the background fill of components. (Inherited from FlexContainer)

Name	Description
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)

Name	Description
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexCartesianChart Class

Description

A base class for the common chart types. CartesianChart defines the basic layout behavior of the standard rectangular, two-dimensional charts.

Inheritance Hierarchy

- [FlexChart](#)
 - FlexCartesianChart
 - [FlexAreaChart](#)
 - [FlexBarChart](#)
 - [FlexBubbleChart](#)
 - [FlexCandlestickChart](#)
 - [FlexColumnChart](#)
 - [FlexHLOCChart](#)
 - [FlexLineChart](#)
 - [FlexPlotChart](#)

Syntax

```
'Declaration
Public Class FlexCartesianChart _
Inherits FlexChart
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipContent	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered. (Inherited from FlexChart)

Name	Description
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexChart</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dataTipMode</i>	how Flex displays DataTip controls for the chart. (Inherited from <i>FlexChart</i>)
<i>description</i>	a short description of the data in the chart. (Inherited from <i>FlexChart</i>)
<i>dragEnabled</i>	whether you can drag items out of this chart and drop them on other controls. (Inherited from <i>FlexChart</i>)
<i>dropEnabled</i>	whether dragged items can be dropped onto the chart. (Inherited from <i>FlexChart</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChart</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexChart</i>)
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11. (Inherited from <i>FlexChart</i>)
<i>gutterBottom</i>	The size of the region, in pixels, between the bottom of the chart data area and the bottom of the chart control.
<i>gutterLeft</i>	The size of the region, in pixels, between the left of the chart data area and the left of the chart control.
<i>gutterRight</i>	The size of the region, in pixels, between the right side of the chart data area and the outside of the chart control.
<i>gutterTop</i>	The size of the region, in pixels, between the top of the chart data area and the top of the chart control.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAxisRatio</i>	the height limit of the horizontal axis.
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChart</i>)
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show. (Inherited from <i>FlexChart</i>)

Name	Description
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>mouseSensitivity</i>	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart. (Inherited from <i>FlexChart</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChart</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChart</i>)
<i>paddingBottom</i>	the number of pixels between the chart's bottom border and its content area. (Inherited from <i>FlexChart</i>)
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from <i>FlexChart</i>)
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from <i>FlexChart</i>)
<i>paddingTop</i>	the number of pixels between the chart's top border and its content area. (Inherited from <i>FlexChart</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedChartItem</i>	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item. (Inherited from <i>FlexChart</i>)
<i>selectedChartItems</i>	an array of all the selected ChartItem objects in the chart. (Inherited from <i>FlexChart</i>)

Name	Description
selectionMode	whether ChartItem objects can be selected. (Inherited from FlexChart)
showDataTips	whether DataTip controls for the chart show. (Inherited from FlexChart)
showDataTipTargets	whether to show targets over the datapoints when showDataTips is set to true. (Inherited from FlexChart)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
textAlign	the alignment of text within a container.
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TextObject)
verticalAxisRatio	the width limit of the vertical axis.
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexChart)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexChart)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexChart)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChart)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header. (Inherited from FlexChart)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 ToString 	Returns a string representation of the control. (Inherited from FlexDisplayObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An

Name	Description
WaitForPropertyNotEquals	exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexChart Class

Description

The link between the Flex component architecture and the DualStyleObject architecture. It extends the Flex UIComponent base class, but contains DualStyleObject classes.

You typically do not use the ChartBase class directly. Instead you use one of its subclasses, such as PlotChart or BubbleChart. It acts as the base class for the common chart types provided in the mx.charts package.

This class defines a number of CSS styles and properties that provide easy access to the more common features of the framework.

A chart's minimum size is 20 x 20 pixels. A chart's maximum size is unbounded. A chart's preferred size is 400 x 400 pixels.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexChart
 - [FlexCartesianChart](#)
 - [FlexPieChart](#)

Syntax

```
'Declaration
Public Class FlexChart _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)

Name	Description
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>clipContent</i>	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered.
<i>columnNames</i>	a list containing the names of all columns in the data.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dataTipMode</i>	how Flex displays DataTip controls for the chart.
<i>description</i>	a short description of the data in the chart.
<i>dragEnabled</i>	whether you can drag items out of this chart and drop them on other controls.
<i>dropEnabled</i>	whether dragged items can be dropped onto the chart.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use.
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child.
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If

Name	Description
	mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
mouseSensitivity	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart.
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available.
numRows	the total number of rows of data available.
paddingBottom	the number of pixels between the chart's bottom border and its content area.
paddingLeft	the number of pixels between the control's left border and the left edge of its content area. The default value is 2.
paddingRight	the number of pixels between the control's right border and the right edge of its content area. The default value is 0.
paddingTop	the number of pixels between the chart's top border and its content area.
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedChartItem	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item.
selectedChartItems	an array of all the selected ChartItem objects in the chart.
selectionMode	whether ChartItem objects can be selected.
showDataTips	whether DataTip controls for the chart show.
showDataTipTargets	whether to show targets over the datapoints when showDataTips is set to true.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event.
DragDrop	Defines the value of the type property of the event object for a dragDrop event.

Name	Description
<i>DragStart</i>	Defines the value of the type property of the event object for a dragStart event.
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from <i>FlexDisplayObject</i>)
<i>GetValues</i>	Returns a matrix containing the automation values of all parts of the components.
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)

Name	Description
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Select</i>	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header.
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>ToString</i>	Returns a string representation of the control. (Inherited from <i>FlexDisplayObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexChartLegend Class

Description

Adds a legend to your charts, where the legend displays the label for each data series in the chart and a key showing the chart element for the series.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexChartLegend

Syntax

```
'Declaration  
Public Class FlexChartLegend _  
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data.
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child.
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
fontWeight	whether the text is boldface.
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
label	the text that displays next to the legend element.
labelPlacement	the label placement of the legend element. Valid values include: <i>top</i> , <i>bottom</i> , <i>right</i> , and <i>left</i> .
lastVisibleRow	the index of the last visible child.

Name	Description
<i>markerAspectRatio</i>	the aspect ratio for the marker associated with this legend item.
<i>markerHeight</i>	the height of the legend element.
<i>markerWidth</i>	the width of the legend element.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available.
<i>numRows</i>	the total number of rows of data available.
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2.
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)

Name	Description
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalGap	the number of pixels between children in the vertical direction. The default value depends on the component class; if not overridden for the class, the default value is 6.
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
LegendItemClick	Dispatched when the user clicks on a LegendItem in the Legend control, which indicates that the user clicked the mouse button over a legend item.

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexChartSeries Class

Description

The base class for the classes that let you specify a data series for a chart control.

You use the subclasses of the Series class with the associated chart control. You can use a Series class to specify the fill pattern and stroke characteristics for the chart elements that are associated with the data series.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexChartSeries
 - [FlexAreaSeries](#)

- [FlexBarSeries](#)
- [FlexBubbleSeries](#)
- [FlexColumnSeries](#)
- [FlexHLOCSeriesBase](#)
- [FlexLineSeries](#)
- [FlexPieSeries](#)
- [FlexPlotSeries](#)

Syntax

```
'Declaration
Public Class FlexChartSeries _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data.
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child.
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
lastVisibleRow	the index of the last visible child.
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject

Name	Description
	instance that is on the display list receives mouse events. If <code>mouseEnabled</code> is set to <code>false</code> , the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the <code>mouseEnabled</code> behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available.
numRows	the total number of rows of data available.
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the <code>width</code> or <code>explicitWidth</code> properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedIndex	the index of the selected item in the data provider of the series.
selectedIndices	an array of indexes of the selected items in the data provider of the series.
selectedItem	the chart item that is selected in the series.
selectedItems	an array of chart items that are selected in the series.
tabChildren	whether the children of the display object are tab enabled. The default is <code>true</code> . (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is <code>Nothing</code> . (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is <code>true</code> . If <code>useHandCursor</code> is <code>false</code> , the arrow pointer is used instead. (Inherited from FlexObject)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from <i>FlexDisplayObject</i>)
<i>GetValues</i>	Returns a matrix containing the automation values of all parts of the components.
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart.
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart.
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexCheckBox Class

Description

The `CheckBox` control consists of an optional label and a small box that can contain a check mark or not.

You can place the optional text label to the left, right, top, or bottom of the `CheckBox`. When a user clicks a `CheckBox` control or its associated text, the `CheckBox` control changes its state from checked to unchecked or from unchecked to checked. `CheckBox` controls gather a set of true or false values that are not mutually exclusive.

Inheritance Hierarchy

- [FlexButton](#)
 - FlexCheckBox

Syntax

```
'Declaration  
Public Class FlexCheckBox _  
Inherits FlexButton
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
color	the color of text in the component, including the component label. The default value is 0x0B333C. (Inherited from FlexButton)
currentState	the current view state of the component. (Inherited from FlexObject)
disabledColor	the color of text in the component if it is disabled. The default value is 0xAAB3B3. (Inherited from FlexButton)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
fillAlphas	the alphas used for the background fill of controls. (Inherited from FlexButton)
fillColors	the colors used to tint the background of the control. (Inherited from FlexButton)
focusAlpha	the alpha transparency value of the focus skin. The default value is 0.4. (Inherited from FlexButton)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
fontFamily	the name of the font to use. (Inherited from FlexButton)
fontSize	the height of the text, in pixels. (Inherited from FlexButton)
fontStyle	whether the text is italic font. (Inherited from FlexButton)
fontWeight	whether the text is boldface. (Inherited from FlexButton)

Name	Description
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>label</i>	the text to appear on the Button control. (Inherited from FlexButton)
<i>labelPlacement</i>	the orientation of the label in relation to a specified icon. (Inherited from FlexButton)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
<i>selected</i>	whether a toggle button is toggled on (true) or off (false). (Inherited from FlexButton)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from FlexDisplayObject)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
<i>Text</i>	The text of the control. (Inherited from TestObject)
<i>textAlign</i>	the alignment of text within a container. (Inherited from FlexButton)
<i>textDecoration</i>	whether the text is underlined. (Inherited from FlexButton)

Name	Description
textIndent	the offset of the first line of text from the left side of the container, in pixels. The default value is 0. (Inherited from FlexButton)
textRollOverColor	the text color of the label as the user moves the mouse pointer over the button. The default value is 0x2B333C. (Inherited from FlexButton)
textSelectedColor	the text color of the label as the user presses it. The default value is 0x000000. (Inherited from FlexButton)
themeColor	the theme color of a component. (Inherited from FlexObject)
toggle	whether a Button is in a toggle state or not. (Inherited from FlexButton)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexButton)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
<i>WaitForObject</i>	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
<i>WaitForProperty</i>	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
<i>WaitForPropertyNotEquals</i>	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
<i>WaitForScreenshotStable</i>	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexColorPicker Class

Description

Provides a way for a user to choose a color from a swatch list.

The default mode of the component shows a single swatch in a square button. When the user clicks the swatch button, the swatch panel appears and displays the entire swatch list.

Inheritance Hierarchy

- [FlexComboBase](#)
 - FlexColorPicker

Syntax

```
'Declaration  
Public Class FlexColorPicker _  
Inherits FlexComboBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. (Inherited from FlexComboBase)
backgroundColor	the background color of a component. (Inherited from FlexComboBase)
backgroundImage	the background image of a component. (Inherited from FlexComboBase)
backgroundSize	the percentage to change the image size to for the specified backgroundImage. (Inherited from FlexComboBase)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
color	the color of text in the component, including the component label. (Inherited from FlexComboBase)
columnCount	the number of columns in the swatch grid. The default value is 20.
currentState	the current view state of the component. (Inherited from FlexObject)
disabledColor	the background color of the component when it is disabled. (Inherited from FlexComboBase)
dropShadowEnabled	whether the ComboBase container's drop shadow is visible. The default value is true. (Inherited from FlexComboBase)
editable	whether the control is editable, which lets the user directly type entries that are not specified in the dataProvider, or not editable, which requires the user select from the items in the dataProvider. (Inherited from FlexComboBase)

Name	Description
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alphas used for the background fill of controls. (Inherited from <i>FlexComboBase</i>)
<i>fillColors</i>	the colors used to tint the background of the control. (Inherited from <i>FlexComboBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexComboBase</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexComboBase</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexComboBase</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexComboBase</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>highlightColor</i>	the highlight color of the drop-down list. (Inherited from <i>FlexComboBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)

Name	Description
<i>restrict</i>	the set of characters that a user can or cannot enter into the text field. (Inherited from <i>FlexComboBase</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedColor</i>	the value of the currently selected color in the SwatchPanel object.
<i>selectedIndex</i>	the index in the data provider of the selected item. (Inherited from <i>FlexComboBase</i>)
<i>showTextField</i>	whether to show the text box that displays the color label or hexadecimal color value. The default value is true.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>text</i>	the contents of the text field. (Inherited from <i>FlexComboBase</i>)
<i>textAlign</i>	the alignment of text within a container. Possible values are:left, right, or center. (Inherited from <i>FlexComboBase</i>)
<i>textDecoration</i>	whether the text is underlined. (Inherited from <i>FlexComboBase</i>)
<i>textIndent</i>	the offset of first line of text from the left side of the container, in pixels. The default value is 0. (Inherited from <i>FlexComboBase</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when the selected color changes as a result of user interaction.
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Defines the value of the type property of the event object for a close event. (Inherited from FlexComboBase)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Input	Creates an Event object that contains information about text events. Event objects are passed as parameters to event listeners. (Inherited from FlexComboBase)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
Open	Defines the value of the type property of the event object for a open event. (Inherited from FlexComboBase)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Defines the value of the type property of the event object for a scroll event. (Inherited from FlexComboBase)
SelectText	Lets you track selection within a text field. (Inherited from FlexComboBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexComboBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change

Name	Description
	the timeout by setting the value of the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexColumnChart Class

Description

Represents data as a series of vertical columns whose height is determined by values in the data. You can use the `ColumnChart` to represent a variety of different charts including simple columns, clustered columns, stacked, 100% stacked, and high/low.

Inheritance Hierarchy

- [FlexCartesianChart](#)
 - FlexColumnChart

Syntax

```
'Declaration
Public Class FlexColumnChart _
Inherits FlexCartesianChart
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>clipContent</i>	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered. (Inherited from <i>FlexChart</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexChart</i>)
<i>columnWidthRatio</i>	a ratio of wide to draw the columns relative to the category width, as a percentage in the range of 0 to 1.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dataTipMode</i>	how Flex displays DataTip controls for the chart. (Inherited from <i>FlexChart</i>)
<i>description</i>	a short description of the data in the chart. (Inherited from <i>FlexChart</i>)
<i>dragEnabled</i>	whether you can drag items out of this chart and drop them on other controls. (Inherited from <i>FlexChart</i>)
<i>dropEnabled</i>	whether dragged items can be dropped onto the chart. (Inherited from <i>FlexChart</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChart</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexChart</i>)
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11. (Inherited from <i>FlexChart</i>)

Name	Description
<i>gutterBottom</i>	The size of the region, in pixels, between the bottom of the chart data area and the bottom of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterLeft</i>	The size of the region, in pixels, between the left of the chart data area and the left of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterRight</i>	The size of the region, in pixels, between the right side of the chart data area and the outside of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterTop</i>	The size of the region, in pixels, between the top of the chart data area and the top of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAxisRatio</i>	the height limit of the horizontal axis. (Inherited from <i>FlexCartesianChart</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChart</i>)
<i>maxColumnWidth</i>	how wide to draw the columns, in pixels.
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show. (Inherited from <i>FlexChart</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>mouseSensitivity</i>	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart. (Inherited from <i>FlexChart</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChart</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChart</i>)
<i>paddingBottom</i>	the number of pixels between the chart's bottom border and its content area. (Inherited from <i>FlexChart</i>)
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from <i>FlexChart</i>)

Name	Description
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from <i>FlexChart</i>)
<i>paddingTop</i>	the number of pixels between the chart's top border and its content area. (Inherited from <i>FlexChart</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedChartItem</i>	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item. (Inherited from <i>FlexChart</i>)
<i>selectedChartItems</i>	an array of all the selected ChartItem objects in the chart. (Inherited from <i>FlexChart</i>)
<i>selectionMode</i>	whether ChartItem objects can be selected. (Inherited from <i>FlexChart</i>)
<i>showDataTips</i>	whether DataTip controls for the chart show. (Inherited from <i>FlexChart</i>)
<i>showDataTipTargets</i>	whether to show targets over the datapoints when showDataTips is set to true. (Inherited from <i>FlexChart</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TextObject</i>)
<i>textAlign</i>	the alignment of text within a container. (Inherited from <i>FlexCartesianChart</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAxisRatio	the width limit of the vertical axis. (Inherited from FlexCartesianChart)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexChart)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexChart)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexChart)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChart)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header. (Inherited from FlexChart)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
WaitForObject	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexColumnSeries Class

Description

Defines a data series for a ColumnChart control.

Inheritance Hierarchy

- [FlexChartSeries](#)
 - FlexColumnSeries

Syntax

```
'Declaration
Public Class FlexColumnSeries _
Inherits FlexChartSeries
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexChartSeries</i>)
<i>columnWidthRatio</i>	the width of columns relative to the category width.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChartSeries</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChartSeries</i>)
<i>maxColumnWidth</i>	the width of the columns, in pixels.
<i>minField</i>	the field of the data provider that determines the y-axis location of the bottom of a column.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any <i>InteractiveObject</i> instance that is on the display list receives mouse events. If <i>mouseEnabled</i> is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the <i>mouseEnabled</i> behavior for all children of an object on the display list, use

Name	Description
	flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available. (Inherited from FlexChartSeries)
numRows	the total number of rows of data available. (Inherited from FlexChartSeries)
offset	how far to offset the center of the columns from the center of the available space, relative to the category width.
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedIndex	the index of the selected item in the data provider of the series. (Inherited from FlexChartSeries)
selectedIndices	an array of indexes of the selected items in the data provider of the series. (Inherited from FlexChartSeries)
selectedItem	the chart item that is selected in the series. (Inherited from FlexChartSeries)
selectedItems	an array of chart items that are selected in the series. (Inherited from FlexChartSeries)
sortOnXField	whether the columns are sorted from left to right before rendering.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)

Name	Description
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>xField</i>	the field of the data provider that determines the x-axis location of the column.
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>yField</i>	the field of the data provider that determines the y-axis location of the top of a column.

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChartSeries)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexComboBase Class

Description

The base class for controls that display text in a text field and have a button that causes a drop-down list to appear where the user can choose which text to display.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexComboBase
 - [FlexColorPicker](#)
 - [FlexComboBox](#)
 - [FlexDateField](#)

Syntax

```
'Declaration  
Public Class FlexComboBase _  
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style.
backgroundColor	the background color of a component.
backgroundImage	the background image of a component.
backgroundSize	the percentage to change the image size to for the specified backgroundImage.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
color	the color of text in the component, including the component label.
currentState	the current view state of the component. (Inherited from FlexObject)
disabledColor	the background color of the component when it is disabled.
dropShadowEnabled	whether the ComboBase container's drop shadow is visible. The default value is true.
editable	whether the control is editable, which lets the user directly type entries that are not specified in the dataProvider, or not editable, which requires the user select from the items in the dataProvider.
enabled	whether the component can accept user interaction. (Inherited from FlexObject)

Name	Description
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alphas used for the background fill of controls.
<i>fillColors</i>	the colors used to tint the background of the control.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use.
<i>fontSize</i>	the height of the text, in pixels.
<i>fontStyle</i>	whether the text is italic font.
<i>fontWeight</i>	whether the text is boldface.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>highlightColor</i>	the highlight color of the drop-down list.
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>restrict</i>	the set of characters that a user can or cannot enter into the text field.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index in the data provider of the selected item.

Name	Description
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>text</i>	the contents of the text field.
<i>textAlign</i>	the alignment of text within a container. Possible values are:left, right, or center.
<i>textDecoration</i>	whether the text is underlined.
<i>textIndent</i>	the offset of first line of text from the left side of the container, in pixels. The default value is 0.
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Defines the value of the type property of the event object for a close event.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Input	Creates an Event object that contains information about text events. Event objects are passed as parameters to event listeners.
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
Open	Defines the value of the type property of the event object for a open event.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Defines the value of the type property of the event object for a scroll event.
SelectText	Lets you track selection within a text field.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexComboBox Class

Description

Contains a drop-down list from which the user can select a single value.

The ComboBox can be editable, in which case the user can type entries into the TextInput portion of the ComboBox that are not in the list.

Inheritance Hierarchy

- [FlexComboBase](#)
 - FlexComboBox

Syntax

```
'Declaration
Public Class FlexComboBox _
Inherits FlexComboBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
alternatingItemColors	the set of BackgroundColors for drop-down list rows in an alternating pattern.
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. (Inherited from FlexComboBase)
backgroundColor	the background color of a component. (Inherited from FlexComboBase)
backgroundImage	the background image of a component. (Inherited from FlexComboBase)

Name	Description
<i>backgroundSize</i>	the percentage to change the image size to for the specified backgroundImage. (Inherited from <i>FlexComboBase</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>color</i>	the color of text in the component, including the component label. (Inherited from <i>FlexComboBase</i>)
<i>columnNames</i>	a list containing the names of all columns in the data.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>disabledColor</i>	the background color of the component when it is disabled. (Inherited from <i>FlexComboBase</i>)
<i>dropShadowEnabled</i>	whether the ComboBase container's drop shadow is visible. The default value is true. (Inherited from <i>FlexComboBase</i>)
<i>editable</i>	whether the control is editable, which lets the user directly type entries that are not specified in the dataProvider, or not editable, which requires the user select from the items in the dataProvider. (Inherited from <i>FlexComboBase</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alphas used for the background fill of controls. (Inherited from <i>FlexComboBase</i>)
<i>fillColors</i>	the colors used to tint the background of the control. (Inherited from <i>FlexComboBase</i>)
<i>firstVisibleRow</i>	the index of the first visible child.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexComboBase</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexComboBase</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexComboBase</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexComboBase</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>highlightColor</i>	the highlight color of the drop-down list. (Inherited from <i>FlexComboBase</i>)

Name	Description
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available.
<i>numRows</i>	the total number of rows of data available.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>restrict</i>	the set of characters that a user can or cannot enter into the text field. (Inherited from <i>FlexComboBase</i>)
<i>rollOverColor</i>	the rollOverColor of the drop-down list.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index in the data provider of the selected item. (Inherited from <i>FlexComboBase</i>)
<i>selectionColor</i>	the selectionColor of the drop-down list.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>text</i>	the contents of the text field. (Inherited from <i>FlexComboBase</i>)
<i>textAlign</i>	the alignment of text within a container. Possible values are:left, right, or center. (Inherited from <i>FlexComboBase</i>)

Name	Description
textDecoration	whether the text is underlined. (Inherited from FlexComboBase)
textIndent	the offset of first line of text from the left side of the container, in pixels. The default value is 0. (Inherited from FlexComboBase)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Defines the value of the type property of the event object for a close event. (Inherited from FlexComboBase)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
<i>Input</i>	Creates an Event object that contains information about text events. Event objects are passed as parameters to event listeners. (Inherited from <i>FlexComboBase</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>Open</i>	Defines the value of the type property of the event object for a open event. (Inherited from <i>FlexComboBase</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Scroll</i>	Defines the value of the type property of the event object for a scroll event. (Inherited from <i>FlexComboBase</i>)
<i>Select</i>	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected.
<i>SelectIndex</i>	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected.
<i>SelectText</i>	Lets you track selection within a text field. (Inherited from <i>FlexComboBase</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexComboBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexContainer Class

Description

An abstract base class for components that controls the layout characteristics of child components.

You do not create an instance of Container in an application. Instead, you create an instance of one of Container's subclasses, such as Canvas or HBox.

The Container class contains the logic for scrolling, clipping, and dynamic instantiation. It contains methods for adding and removing children and the logic for drawing the background and borders of containers.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexContainer
 - [FlexAccordion](#)
 - [FlexApplication](#)
 - [FlexBox](#)
 - [FlexCanvas](#)
 - [FlexForm](#)
 - [FlexFormItem](#)
 - [FlexPanel](#)
 - [FlexViewStack](#)
 - [FlexWindow](#)

Syntax

```
'Declaration
Public Class FlexContainer _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)

Name	Description
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>backgroundAlpha</i>	the alpha level of the color defined by the <i>backgroundColor</i> property, of the image or SWF file defined by the <i>backgroundImage</i> style. Valid values range from 0.0 to 1.0.
<i>backgroundColor</i>	the background color of a component.
<i>backgroundDisabledColor</i>	the background color of the component when it is disabled. The global default value is undefined.
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class.
<i>backgroundSize</i>	the percentage to change the image size to for the specified <i>backgroundImage</i> .
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data.
<i>creationIndex</i>	the order to instantiate and draw the children of the container.
<i>creationPolicy</i>	the child creation policy for this Container.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components.
<i>fillColors</i>	the colors used to tint the background fill of the component.
<i>firstVisibleRow</i>	the index of the first visible child.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use.
<i>fontSize</i>	the height of the text, in pixels.
<i>fontStyle</i>	whether the text is italic font.
<i>fontWeight</i>	whether the text is boldface.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container.
<i>lastVisibleRow</i>	the index of the last visible child.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container.
<i>numColumns</i>	the total number of columns of data available.
<i>numRows</i>	the total number of rows of data available.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)

Name	Description
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>DragCancel</i>	Defines the value of the type property of the event object for a dragComplete event.
<i>DragDrop</i>	Defines the value of the type property of the event object for a dragDrop event.
<i>DragStart</i>	Defines the value of the type property of the event object for a dragStart event.
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance.
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
<i>WaitForObject</i>	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
<i>WaitForProperty</i>	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
<i>WaitForPropertyNotEquals</i>	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
<i>WaitForScreenshotStable</i>	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexContainerMovieClip Class

Description

Container components created in Adobe Flash CS3 Professional for use in Flex are subclasses of the `mx.flash.ContainerMovieClip` class.

You can use a subclass of `ContainerMovieClip` as a Flex container, it can hold children, and it can respond to events, define view states and transitions, and work with effects in the same way as can any Flex component.

A Flash container can only have a single Flex child. However, this child can be a Flex container which allows you to add additional children.

Inheritance Hierarchy

- [FlexUIMovieClip](#)
 - FlexContainerMovieClip

Syntax

```
'Declaration  
Public Class FlexContainerMovieClip _  
Inherits FlexUIMovieClip
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexUIMovieClip)
automationIndex	a string containing the automation index for the element. (Inherited from FlexUIMovieClip)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexUIMovieClip)
className	the name of this instance's class, such as "Button". (Inherited from FlexUIMovieClip)
currentState	the current state of this component. (Inherited from FlexUIMovieClip)
enabled	whether a movie clip is enabled. (Inherited from FlexUIMovieClip)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexUIMovieClip)
errorString	the text that is displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexUIMovieClip)
focusEnabled	whether the component can receive focus when selected. (Inherited from FlexUIMovieClip)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexUIMovieClip)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)

Name	Description
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexUIMovieClip</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. Valid values are 0 to 100. (Inherited from <i>FlexUIMovieClip</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Valid values are 0 to 100. (Inherited from <i>FlexUIMovieClip</i>)
<i>scaleX</i>	the horizontal scale (percentage) of the object as applied from the registration point. The default registration point is (0,0). 1.0 equals 100% scale. (Inherited from <i>FlexUIMovieClip</i>)
<i>scaleY</i>	the vertical scale (percentage) of an object as applied from the registration point of the object. The default registration point is (0,0). 1.0 is 100% scale. (Inherited from <i>FlexUIMovieClip</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexUIMovieClip</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexUIMovieClip</i>)
<i>useHandCursor</i>	whether the pointing hand (hand cursor) appears when the mouse rolls over a sprite in which the buttonMode property is set to true. (Inherited from <i>FlexUIMovieClip</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Defines the value of the type property of a keyFocusChange event object. (Inherited from FlexUIMovieClip)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragDrop	Defines the value of the type property of the event object for a dragDrop event.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Called by the FocusManager when the component receives focus. The component may in turn set focus to an internal component. (Inherited from FlexUIMovieClip)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexDataGrid Class

Description

The DataGrid control is like a List except that it can show more than one column of data making it suited for showing objects with multiple properties.

Inheritance Hierarchy

- [FlexListBase](#)
 - FlexDataGrid

Syntax

```
'Declaration
Public Class FlexDataGrid _
Inherits FlexListBase
```

Properties

Name	Description
allowMultipleSelection	whether you can allow more than one item to be selected at the same time. (Inherited from FlexListBase)
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
alternatingItemColors	the colors to use for the backgrounds of the items in the list. (Inherited from FlexListBase)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundDisabledColor	the color of text in the component if it is disabled. The default value is 0xAAB3B3. (Inherited from FlexListBase)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnCount	the number of columns to be displayed in a TileList control or items in a HorizontalList control. (Inherited from FlexListBase)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexListBase)

Name	Description
<i>columnWidth</i>	the width of the control's columns. (Inherited from <i>FlexListBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dataTipField</i>	the name of the field in the data provider items to display as the data tip. (Inherited from <i>FlexListBase</i>)
<i>editable</i>	whether the user can edit items in the data provider.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexListBase</i>)
<i>focusAlpha</i>	the alpha transparency value of the focus skin. The default value is 0.4. (Inherited from <i>FlexListBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>headerColors</i>	An array of two colors used to draw the header background gradient.
<i>headerStyleName</i>	the name of a CSS style declaration for controlling other aspects of the appearance of the column headers. The default value is <i>dataGridStyles</i> .
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalGridLineColor</i>	the color of the horizontal grid lines.
<i>horizontalGridLines</i>	whether to show horizontal grid lines between the rows.
<i>iconField</i>	the name of the field in the data provider object that determines what to display as the icon. (Inherited from <i>FlexListBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>labelField</i>	the name of the field in the data provider items to display as the label. (Inherited from <i>FlexListBase</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexListBase</i>)
<i>lockedColumnCount</i>	the index of the first column in the control that scrolls. (Inherited from <i>FlexListBase</i>)
<i>lockedRowCount</i>	the index of the first row in the control that scrolls. (Inherited from <i>FlexListBase</i>)
<i>minColumnWidth</i>	The minimum width of the columns, in pixels. If not NaN, the DataGrid control applies this value as the minimum width for all

Name	Description
	columns. Otherwise, individual columns can have their own minimum widths. The default value is NaN.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexListBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexListBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizableColumns</i>	whether the user can change the size of the columns.
<i>rollOverColor</i>	the color of the background of a renderer when the user rolls over it. The default value is 0xEEFEE6. (Inherited from <i>FlexListBase</i>)
<i>rowCount</i>	the number of rows to be displayed. (Inherited from <i>FlexListBase</i>)
<i>rowHeight</i>	the height of the rows in pixels. (Inherited from <i>FlexListBase</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectable</i>	whether the list shows selected items as selected. (Inherited from <i>FlexListBase</i>)
<i>selectedIndex</i>	the index in the data provider of the selected item. (Inherited from <i>FlexListBase</i>)
<i>selectedIndices</i>	an array of indices in the data provider of the selected items. (Inherited from <i>FlexListBase</i>)
<i>selectedItem</i>	a reference to the selected item in the data provider. (Inherited from <i>FlexListBase</i>)

Name	Description
<i>selectedItems</i>	an array of references to the selected items in the data provider. (Inherited from <i>FlexListBase</i>)
<i>selectionColor</i>	the color of the background of a renderer when the user selects it. The default value is 0x7FCEFF. (Inherited from <i>FlexListBase</i>)
<i>selectionDisabledColor</i>	the color of the background of a renderer when the component is disabled. The default value is 0xDDDDDD. (Inherited from <i>FlexListBase</i>)
<i>showDataTips</i>	whether dataTips are displayed for text in the rows. (Inherited from <i>FlexListBase</i>)
<i>sortableColumns</i>	whether the user can sort the data provider items by clicking on a column header cell.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textRollOverColor</i>	the text color of the label as the user moves the mouse pointer over the button. The default value is 0x2B333C. (Inherited from <i>FlexListBase</i>)
<i>textSelectedColor</i>	the color of the text of a renderer when the user selects it. The default value is 0x2B333C. (Inherited from <i>FlexListBase</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>useRollOver</i>	whether items are highlighted as the mouse rolls over them. (Inherited from <i>FlexListBase</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>variableRowHeight</i>	whether the individual rows can have different height. (Inherited from <i>FlexListBase</i>)
<i>verticalAlign</i>	the vertical alignment of a renderer in a row. (Inherited from <i>FlexListBase</i>)
<i>verticalGridLineColor</i>	the color of the vertical grid lines. The default value is 0x666666.
<i>verticalGridLines</i>	whether to show vertical grid lines between the columns.

Name	Description
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
wordWrap	whether text in the row should be word wrapped. (Inherited from FlexListBase)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
ColumnStretch	Dispatched when a user changes the width of a column, indicating that the amount of data displayed in that column may have changed.
Deselect	Defines the value of the type property of the event object for an event that is dispatched when a previously selected item is deselected. (Inherited from FlexListBase)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Defines the value of the type property of the ListEvent object for an itemDoubleClick event, which indicates that the user double-clicked the mouse over a visual item in the control. (Inherited from FlexListBase)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexListBase)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexListBase)

Name	Description
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexListBase)
Edit	Defines the value of the type property of the event object for a itemEditBegin event, which indicates that an item is ready to be edited.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexListBase)
HeaderClick	Dispatched when the user releases the mouse button on a column header to request the control to sort the grid contents based on the contents of the column.
HeaderShift	Dispatched when the user releases the mouse button on a column header after having dragged the column to a new location resulting in shifting the column to a new index.
HighlightObject	Highlights this object. (Inherited from TestObject)

Name	Description
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexScrollBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
MultiSelect	Defines the value of the type property of the event object for an event that is dispatched when an item is selected as part of an action that selects multiple items. (Inherited from FlexListBase)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Defines the value of the type property of the event object for a scroll event. (Inherited from FlexListBase)
ScrollToIndex	Ensures that the data provider item at the given index is visible. (Inherited from FlexListBase)
Select	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected. (Inherited from FlexListBase)

Name	Description
SelectIndex	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected. (Inherited from FlexListBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexListBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexDateChooser Class

Description

Displays the name of a month, the year, and a grid of the days of the month, with columns labeled for the day of the week.

The user can select a date, a range of dates, or multiple dates. The control contains forward and back arrow buttons for changing the month and year.

You can let users select multiple dates, disable the selection of certain dates, and limit the display to a range of dates.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexDateChooser

Syntax

```
'Declaration
Public Class FlexDateChooser _
Inherits FlexObject
```

Properties

Name	Description
<i>allowDisjointSelection</i>	whether non-contiguous selection is allowed in the DateChooser control.
<i>allowMultipleSelection</i>	whether multiple selection is allowed in the DateChooser control.
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>color</i>	the color of text in the component, including the component label. The default value is 0x0B333C.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>disabledColor</i>	the color of text in the component if it is disabled. The default value is 0xAAB3B3.
<i>disabledDays</i>	the days to disable in a week.
<i>disabledRanges</i>	the single and multiple days to disable.
<i>displayedMonth</i>	the month displayed in the DateChooser control.
<i>displayedYear</i>	the year displayed in the DateChooser control. The default value is the current year.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillColors</i>	the colors used to tint the background of the control.
<i>firstDayOfWeek</i>	the number representing the day of the week to display in the first column of the DateChooser control.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use.
<i>fontSize</i>	the height of the text, in pixels.
<i>fontStyle</i>	whether the text is italic font.

Name	Description
<i>fontWeight</i>	whether the text is boldface.
<i>headerColors</i>	the colors of the band at the top of the DateChooser control. The default value is [0xE6EEEE, 0xFFFFFFFF].
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>maxYear</i>	the last year selectable in the control.
<i>minYear</i>	the first year selectable in the control.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>rollOverColor</i>	the color of the highlight area of the date when the user holds the mouse pointer over a date in the DateChooser control. The default value is 0xE3FFD6.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedDate</i>	the date as selected in the DateChooser control.
<i>selectedRanges</i>	the selected date ranges.
<i>selectionColor</i>	the color of the highlight area of the currently selected date in the DateChooser control. The default value is 0xCDFFC1.
<i>showToday</i>	whether today's date is highlighted in the DataChooser control.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
textAlign	whether the text is boldface.
textDecoration	whether the text is underlined.
textIndent	the offset of first line of text from the left side of the container, in pixels. The default value is 0.
themeColor	the theme color of a component. (Inherited from FlexObject)
todayColor	the color of the highlight of today's date in the DateChooser control. The default value is 0x2B333.
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
yearNavigationEnabled	whether year navigation is enabled.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when a date is selected or changed.

Name	Description
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollDate	The DateChooserEvent.SCROLL constant defines the value of the type property of the event object for a scrollevent.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexDateField Class

Description

A text field that shows the date with a calendar icon on its right side.

When the user clicks anywhere inside the bounding box of the control, a DateChooser control pops up and shows the dates in the month of the current date. If no date is selected, the text field is blank and the month of the current date is displayed in the DateChooser control.

Inheritance Hierarchy

- [FlexComboBase](#)
 - FlexDateField

Syntax

```
'Declaration
Public Class FlexDateField _
Inherits FlexComboBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. (Inherited from FlexComboBase)
backgroundColor	the background color of a component. (Inherited from FlexComboBase)
backgroundImage	the background image of a component. (Inherited from FlexComboBase)
backgroundSize	the percentage to change the image size to for the specified backgroundImage. (Inherited from FlexComboBase)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
color	the color of text in the component, including the component label. (Inherited from FlexComboBase)
currentState	the current view state of the component. (Inherited from FlexObject)
disabledColor	the background color of the component when it is disabled. (Inherited from FlexComboBase)
disabledDays	the days to disable in a week.
disabledRanges	the disabled single and multiple days.
displayedMonth	the month that is displayed in the DateChooser control.

Name	Description
<i>displayedYear</i>	the year that is displayed in the DateChooser control.
<i>dropShadowEnabled</i>	whether the ComboBase container's drop shadow is visible. The default value is true. (Inherited from <i>FlexComboBase</i>)
<i>editable</i>	whether the control is editable, which lets the user directly type entries that are not specified in the dataProvider, or not editable, which requires the user select from the items in the dataProvider. (Inherited from <i>FlexComboBase</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alphas used for the background fill of controls. (Inherited from <i>FlexComboBase</i>)
<i>fillColors</i>	the colors used to tint the background of the control. (Inherited from <i>FlexComboBase</i>)
<i>firstDayOfWeek</i>	the day of the week (0-6, where 0 is the first element of the dayNames Array) to display in the first column of the DateChooser control.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexComboBase</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexComboBase</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexComboBase</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexComboBase</i>)
<i>headerColors</i>	the colors of the band at the top of the DateChooser control. The default value is [0xE6EEEE, 0xFFFFFFFF].
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>highlightColor</i>	the highlight color of the drop-down list. (Inherited from <i>FlexComboBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>maxYear</i>	the last year selectable in the control.
<i>minYear</i>	the first year selectable in the control.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any

Name	Description
	mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
restrict	the set of characters that a user can or cannot enter into the text field. (Inherited from FlexComboBase)
rollOverColor	the color of the highlight area of the date when the user holds the mouse pointer over a date in the DateChooser control. The default value is 0xE3FFD6.
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedDate	the date as selected in the DateChooser control.
selectedIndex	the index in the data provider of the selected item. (Inherited from FlexComboBase)
showToday	the day highlighted in the DateChooser control.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
text	the contents of the text field. (Inherited from FlexComboBase)
textAlign	the alignment of text within a container. Possible values are:left, right, or center. (Inherited from FlexComboBase)
textDecoration	whether the text is underlined. (Inherited from FlexComboBase)
textIndent	the offset of first line of text from the left side of the container, in pixels. The default value is 0. (Inherited from FlexComboBase)
themeColor	the theme color of a component. (Inherited from FlexObject)

Name	Description
todayColor	the color of the highlight of today's date in the DateChooser control. The default value is 0x2B333.
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
yearNavigationEnabled	whether the year up and down buttons are available for navigation.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when a date is selected or changed, and the DateChooser control closes.
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Defines the value of the type property of the event object for a close event. (Inherited from FlexComboBase)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
<i>Input</i>	Creates an Event object that contains information about text events. Event objects are passed as parameters to event listeners. (Inherited from <i>FlexComboBase</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>Open</i>	Defines the value of the type property of the event object for a open event. (Inherited from <i>FlexComboBase</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Scroll</i>	Defines the value of the type property of the event object for a scroll event. (Inherited from <i>FlexComboBase</i>)
<i>ScrollDate</i>	The DateChooserEvent.SCROLL constant defines the value of the type property of the event object for a scrollevent.
<i>SelectText</i>	Lets you track selection within a text field. (Inherited from <i>FlexComboBase</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexComboBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexDisplayObject Class

Description

The base class for all objects that can be placed on the display list. The display list manages all objects displayed in Flash Player or Adobe AIR.

Inheritance Hierarchy

- [TestObject](#)
 - FlexDisplayObject
 - [FlexListLabel](#)
 - [FlexObject](#)
 - [FlexUIMovieClip](#)

Syntax

```
'Declaration
Public Class FlexDisplayObject _
Inherits TestObject _
Implements IClickable, IKeyable
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified.
height	the height of the display object, in pixels.
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren.
tabChildren	whether the children of the display object are tab enabled. The default is true.
tabEnabled	whether this object is in the tab order.
tabIndex	the tab ordering of objects in a SWF file.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
visible	whether the display object is visible.
width	the width of the display object, in pixels.
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer.
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation.
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified

Name	Description
	interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexDividedBox Class

Description

Measures and lays out its children horizontally or vertically in exactly the same way as a `Box` container, but it inserts draggable dividers in the gaps between the children. Users can drag any divider to resize the children on each side.

The `DividedBox` class is the base class for the `HDividedBox` and `VDividedBox` classes.

Inheritance Hierarchy

- [FlexBox](#)
 - FlexDividedBox

Syntax

```
'Declaration
Public Class FlexDividedBox _
Inherits FlexBox
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the <code>backgroundColor</code> property, of the image or SWF file defined by the <code>backgroundImage</code> style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)
backgroundDisabledColor	the background color of the component when it is disabled. The global default value is undefined. (Inherited from FlexContainer)

Name	Description
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class. (Inherited from <i>FlexContainer</i>)
<i>backgroundSize</i>	the percentage to change the image size to for the specified backgroundImage. (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>direction</i>	whether to use vertical (default) or horizontal layout. (Inherited from <i>FlexBox</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAlign</i>	the horizontal alignment of children in the container. (Inherited from <i>FlexBox</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)

Name	Description
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizeToContent</i>	whether the DividedBox automatically resizes to the size of its children.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)

Name	Description
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAlign	the vertical alignment of children in the container. (Inherited from FlexBox)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
Dragged	Dispatched multiple times as the user drags any divider. The dividerDrag event is dispatched after the dividerPress event and before the dividerRelease event.

Name	Description
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
Pressed	Dispatched when the user presses any divider in this container.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Released	Dispatched when the user releases a divider.
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexForm Class

Description

Use this class to control the layout of a form, mark form fields as required or optional, handle error messages, and bind your form data to the Flex data model to perform data checking and validation. This class also lets you use style sheets to configure the appearance of your forms.

Inheritance Hierarchy

- [FlexContainer](#)
 - FlexForm

Syntax

```
'Declaration
Public Class FlexForm _
Inherits FlexContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)
backgroundDisabledColor	the background color of the component when it is disabled. The global default value is undefined. (Inherited from FlexContainer)
backgroundImage	the background image of a component. This can be an absolute or relative URL or class. (Inherited from FlexContainer)
backgroundSize	the percentage to change the image size to for the specified backgroundImage. (Inherited from FlexContainer)

Name	Description
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all

Name	Description
	children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numChildren	the number of child components in this container. (Inherited from FlexContainer)
numColumns	the total number of columns of data available. (Inherited from FlexContainer)
numRows	the total number of rows of data available. (Inherited from FlexContainer)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or <code>explicitWidth</code> properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If <code>useHandCursor</code> is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)

Name	Description
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>DragCancel</i>	Defines the value of the type property of the event object for a dragComplete event. (Inherited from <i>FlexContainer</i>)
<i>DragDrop</i>	Defines the value of the type property of the event object for a dragDrop event. (Inherited from <i>FlexContainer</i>)
<i>DragStart</i>	Defines the value of the type property of the event object for a dragStart event. (Inherited from <i>FlexContainer</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)

Name	Description
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException

Name	Description
	is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <i>WaitForObject</i> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <i>WaitForObject</i> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <i>WaitForObject</i> statements to your scripts. (Inherited from <i>TestObject</i>)
<i>WaitForProperty</i>	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForPropertyNotEquals</i>	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForScreenshotStable</i>	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

FlexFormItem Class

Description

Defines a label and one or more children arranged horizontally or vertically. Children can be controls or other containers. A single Form container can hold multiple FormItem containers.

Inheritance Hierarchy

- *FlexContainer*
 - FlexFormItem

Syntax

```
'Declaration
Public Class FlexFormItem _
Inherits FlexContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)
backgroundDisabledColor	the background color of the component when it is disabled. The global default value is undefined. (Inherited from FlexContainer)
backgroundImage	the background image of a component. This can be an absolute or relative URL or class. (Inherited from FlexContainer)
backgroundSize	the percentage to change the image size to for the specified backgroundImage . (Inherited from FlexContainer)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexContainer)
creationIndex	the order to instantiate and draw the children of the container. (Inherited from FlexContainer)
creationPolicy	the child creation policy for this Container. (Inherited from FlexContainer)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
fillAlphas	the alpha transparency values used for the background fill of components. (Inherited from FlexContainer)

Name	Description
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAlign</i>	the horizontal alignment of children in the container.
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)

Name	Description
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TextObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TextObject)

Name	Description
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexHLOCChart Class

Description

The HLOCChart (High Low Open Close) control represents financial data as a series of elements representing the high, low, closing, and optionally opening values of a data series.

The top and bottom of the vertical line in each element represent the high and low values for the datapoint. The right-facing tick represents the closing values, and the left tick represents the opening value if one was specified.

Inheritance Hierarchy

- [FlexCartesianChart](#)
 - FlexHLOCChart

Syntax

```
'Declaration
Public Class FlexHLOCChart _
Inherits FlexCartesianChart
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipContent	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered. (Inherited from FlexChart)

Name	Description
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexChart</i>)
<i>columnWidthRatio</i>	a ratio of how wide to draw the HLOC lines relative to the horizontal axis's category widths, as a percentage in the range of 0 to 1.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dataTipMode</i>	how Flex displays DataTip controls for the chart. (Inherited from <i>FlexChart</i>)
<i>description</i>	a short description of the data in the chart. (Inherited from <i>FlexChart</i>)
<i>dragEnabled</i>	whether you can drag items out of this chart and drop them on other controls. (Inherited from <i>FlexChart</i>)
<i>dropEnabled</i>	whether dragged items can be dropped onto the chart. (Inherited from <i>FlexChart</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChart</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexChart</i>)
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11. (Inherited from <i>FlexChart</i>)
<i>gutterBottom</i>	The size of the region, in pixels, between the bottom of the chart data area and the bottom of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterLeft</i>	The size of the region, in pixels, between the left of the chart data area and the left of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterRight</i>	The size of the region, in pixels, between the right side of the chart data area and the outside of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterTop</i>	The size of the region, in pixels, between the top of the chart data area and the top of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>horizontalAxisRatio</i>	the height limit of the horizontal axis. (Inherited from <i>FlexCartesianChart</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChart</i>)
<i>maxColumnWidth</i>	how wide to draw the HLOC lines, in pixels.
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show. (Inherited from <i>FlexChart</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>mouseSensitivity</i>	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart. (Inherited from <i>FlexChart</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChart</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChart</i>)
<i>paddingBottom</i>	the number of pixels between the chart's bottom border and its content area. (Inherited from <i>FlexChart</i>)
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from <i>FlexChart</i>)
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from <i>FlexChart</i>)
<i>paddingTop</i>	the number of pixels between the chart's top border and its content area. (Inherited from <i>FlexChart</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)

Name	Description
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedChartItem	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item. (Inherited from FlexChart)
selectedChartItems	an array of all the selected ChartItem objects in the chart. (Inherited from FlexChart)
selectionMode	whether ChartItem objects can be selected. (Inherited from FlexChart)
showDataTips	whether DataTip controls for the chart show. (Inherited from FlexChart)
showDataTipTargets	whether to show targets over the datapoints when showDataTips is set to true. (Inherited from FlexChart)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
textAlign	the alignment of text within a container. (Inherited from FlexCartesianChart)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAxisRatio	the width limit of the vertical axis. (Inherited from FlexCartesianChart)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Name	Description
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexChart)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexChart)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexChart)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChart)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header. (Inherited from FlexChart)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> ,

Name	Description
WaitForObject	because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexHLOCSeries Class

Description

Represents financial data as a series of elements representing the high, low, closing, and, optionally, opening values of a data series.

The top and bottom of the vertical line in each element represent the high and low values for the datapoint. The right-facing tick mark represents the closing value, and the left tick mark represents the opening value, if one was specified.

Inheritance Hierarchy

- [FlexHLOCSeriesBase](#)
 - FlexHLOCSeries

Syntax

```
'Declaration
Public Class FlexHLOCSeries _
Inherits FlexHLOCSeriesBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)

Name	Description
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexChartSeries</i>)
<i>columnWidthRatio</i>	the width of elements relative to the category width. (Inherited from <i>FlexHLOCSeriesBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChartSeries</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChartSeries</i>)
<i>maxColumnWidth</i>	the width of the elements, in pixels. (Inherited from <i>FlexHLOCSeriesBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)

Name	Description
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChartSeries</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChartSeries</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index of the selected item in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedIndices</i>	an array of indexes of the selected items in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedItem</i>	the chart item that is selected in the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedItems</i>	an array of chart items that are selected in the series. (Inherited from <i>FlexChartSeries</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>xField</i>	the field of the data provider that determines the x-axis location of the element. (Inherited from <i>FlexHLOCSeriesBase</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)

Name	Description
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
<i>GetValues</i>	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChartSeries)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException

Name	Description
	is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexHLOCSeriesBase Class

Description

Represents financial data as a series of elements representing the high, low, closing, and, optionally, opening values of a data series.

The top and bottom of the vertical line in each element represent the high and low values for the datapoint. The right-facing tick mark represents the closing value, and the left tick mark represents the opening value, if one was specified.

Inheritance Hierarchy

- [FlexChartSeries](#)
 - FlexHLOCSeriesBase
 - [FlexCandlestickSeries](#)
 - [FlexHLOCSeries](#)

Syntax

```
'Declaration  
Public Class FlexHLOCSeriesBase _  
Inherits FlexChartSeries
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChartSeries)
columnWidthRatio	the width of elements relative to the category width.
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from FlexChartSeries)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
lastVisibleRow	the index of the last visible child. (Inherited from FlexChartSeries)
maxColumnWidth	the width of the elements, in pixels.
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list

Name	Description
	are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available. (Inherited from FlexChartSeries)
numRows	the total number of rows of data available. (Inherited from FlexChartSeries)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedIndex	the index of the selected item in the data provider of the series. (Inherited from FlexChartSeries)
selectedIndices	an array of indexes of the selected items in the data provider of the series. (Inherited from FlexChartSeries)
selectedItem	the chart item that is selected in the series. (Inherited from FlexChartSeries)
selectedItems	an array of chart items that are selected in the series. (Inherited from FlexChartSeries)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
xField	the field of the data provider that determines the x-axis location of the element.
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChartSeries)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
WaitForObject	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexImage Class

Description

Lets you import JPEG, PNG, GIF, and SWF files at runtime.

You can also embed any of these files and SVG files at compile time by using `@Embed(source='filename')`.

Note: Flex also includes the `SWFLoader` control for loading Flex applications. You typically use the `Image` control for loading static graphic files and SWF files, and use the `SWFLoader` control for loading Flex applications. The `Image` control is also designed to be used in custom item renderers and item editors.

Inheritance Hierarchy

- [FlexLoader](#)
 - FlexImage

Syntax

```
'Declaration  
Public Class FlexImage _  
Inherits FlexLoader
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)

Name	Description
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>source</i>	The URL, object, class or string name of a class to load as the content. (Inherited from <i>FlexLoader</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexLabel Class

Description

Displays a single line of noneditable text.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexLabel

Syntax

```
'Declaration
Public Class FlexLabel _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
color	the color of text in the component, including the component label. The default value is 0x0B333C.
currentState	the current view state of the component. (Inherited from FlexObject)
disabledColor	the color of text in the component if it is disabled. The default value is 0xAAB3B3.
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)

Name	Description
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use.
<i>fontSize</i>	the height of the text, in pixels.
<i>fontStyle</i>	whether the text is italic font.
<i>fontWeight</i>	whether the text is boldface.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>htmlText</i>	the text displayed by the Label control, including HTML markup that expresses the styles of that text.
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)

Name	Description
text	the plain text displayed by this control. Its appearance is determined by the CSS styles of this Label control.
textAlign	the alignment of text within a container.
textDecoration	whether the text is underlined.
textIndent	the offset of the first line of text from the left side of the container, in pixels. The default value is 0.
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexLineChart Class

Description

Represents a data series as points connected by a continuous line.

You can use an icon or symbol to represent each data point along the line, or show a simple line with no icons.

Inheritance Hierarchy

- [FlexCartesianChart](#)
 - FlexLineChart

Syntax

```
'Declaration  
Public Class FlexLineChart _  
Inherits FlexCartesianChart
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipContent	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered. (Inherited from FlexChart)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChart)
currentState	the current view state of the component. (Inherited from FlexObject)
dataTipMode	how Flex displays DataTip controls for the chart. (Inherited from FlexChart)
description	a short description of the data in the chart. (Inherited from FlexChart)
dragEnabled	whether you can drag items out of this chart and drop them on other controls. (Inherited from FlexChart)
dropEnabled	whether dragged items can be dropped onto the chart. (Inherited from FlexChart)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from FlexChart)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
fontFamily	the name of the font to use. (Inherited from FlexChart)

Name	Description
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11. (Inherited from FlexChart)
<i>gutterBottom</i>	The size of the region, in pixels, between the bottom of the chart data area and the bottom of the chart control. (Inherited from FlexCartesianChart)
<i>gutterLeft</i>	The size of the region, in pixels, between the left of the chart data area and the left of the chart control. (Inherited from FlexCartesianChart)
<i>gutterRight</i>	The size of the region, in pixels, between the right side of the chart data area and the outside of the chart control. (Inherited from FlexCartesianChart)
<i>gutterTop</i>	The size of the region, in pixels, between the top of the chart data area and the top of the chart control. (Inherited from FlexCartesianChart)
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>horizontalAxisRatio</i>	the height limit of the horizontal axis. (Inherited from FlexCartesianChart)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from FlexChart)
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show. (Inherited from FlexChart)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
<i>mouseSensitivity</i>	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart. (Inherited from FlexChart)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from FlexChart)
<i>numRows</i>	the total number of rows of data available. (Inherited from FlexChart)
<i>paddingBottom</i>	the number of pixels between the chart's bottom border and its content area. (Inherited from FlexChart)

Name	Description
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from <i>FlexChart</i>)
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from <i>FlexChart</i>)
<i>paddingTop</i>	the number of pixels between the chart's top border and its content area. (Inherited from <i>FlexChart</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedChartItem</i>	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item. (Inherited from <i>FlexChart</i>)
<i>selectedChartItems</i>	an array of all the selected ChartItem objects in the chart. (Inherited from <i>FlexChart</i>)
<i>selectionMode</i>	whether ChartItem objects can be selected. (Inherited from <i>FlexChart</i>)
<i>showDataTips</i>	whether DataTip controls for the chart show. (Inherited from <i>FlexChart</i>)
<i>showDataTipTargets</i>	whether to show targets over the datapoints when showDataTips is set to true. (Inherited from <i>FlexChart</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textAlign</i>	the alignment of text within a container. (Inherited from <i>FlexCartesianChart</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)

Name	Description
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAxisRatio	the width limit of the vertical axis. (Inherited from FlexCartesianChart)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexChart)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexChart)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexChart)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChart)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header. (Inherited from FlexChart)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexLineSeries Class

Description

Defines a data series for a LineChart control.

Inheritance Hierarchy

- [FlexChartSeries](#)
 - FlexLineSeries

Syntax

```
'Declaration  
Public Class FlexLineSeries _  
Inherits FlexChartSeries
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChartSeries)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from FlexChartSeries)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
form	the line type for the chart.
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
lastVisibleRow	the index of the last visible child. (Inherited from FlexChartSeries)

Name	Description
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChartSeries</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChartSeries</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>radius</i>	the radius, in pixels, of the chart elements for the data points.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index of the selected item in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedIndices</i>	an array of indexes of the selected items in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedItem</i>	the chart item that is selected in the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedItems</i>	an array of chart items that are selected in the series. (Inherited from <i>FlexChartSeries</i>)
<i>sortOnXField</i>	whether the line datapoints are sorted from left to right before rendering.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
xField	the field of the data provider that determines the x-axis location of each data point.
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
yField	the field of the data provider that determines the y-axis location of each data point.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is

Name	Description
	passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChartSeries)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexLinkBar Class

Description

Defines a horizontal or vertical row of LinkButton controls that designate a series of link destinations.

You typically use a LinkBar control to control the active child container of a ViewStack container, or to create a stand-alone set of links.

Inheritance Hierarchy

- [FlexNavigationBar](#)
 - FlexLinkBar

Syntax

```
'Declaration  
Public Class FlexLinkBar _  
Inherits FlexNavigationBar
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)
backgroundDisabledColor	the background color of the component when it is disabled. The global default value is undefined. (Inherited from FlexContainer)
backgroundImage	the background image of a component. This can be an absolute or relative URL or class. (Inherited from FlexContainer)
backgroundSize	the percentage to change the image size to for the specified backgroundImage. (Inherited from FlexContainer)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)

Name	Description
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>direction</i>	whether to use vertical (default) or horizontal layout. (Inherited from <i>FlexBox</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAlign</i>	the horizontal alignment of children in the container. (Inherited from <i>FlexBox</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If

Name	Description
	mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numChildren	the number of child components in this container. (Inherited from FlexContainer)
numColumns	the total number of columns of data available. (Inherited from FlexContainer)
numRows	the total number of rows of data available. (Inherited from FlexContainer)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedIndex	the index of the active navigation item, where the first item is at an index of 0. The default value is -1. (Inherited from FlexNavigationBar)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAlign	the vertical alignment of children in the container. (Inherited from FlexBox)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Defines the value of the type property of the event object for an itemClick event. (Inherited from FlexNavigationBar)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)

Name	Description
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexNavigationBar)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexList Class

Description

Displays a list of items horizontally, verically, or laid out in tiles.

Inheritance Hierarchy

- [FlexListBase](#)
 - FlexList
 - [FlexTree](#)

Syntax

```
'Declaration  
Public Class FlexList _  
Inherits FlexListBase
```

Properties

Name	Description
allowMultipleSelection	whether you can allow more than one item to be selected at the same time. (Inherited from FlexListBase)
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
alternatingItemColors	the colors to use for the backgrounds of the items in the list. (Inherited from FlexListBase)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundDisabledColor	the color of text in the component if it is disabled. The default value is 0xAAB3B3. (Inherited from FlexListBase)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnCount	the number of columns to be displayed in a <i>TileList</i> control or items in a <i>HorizontalList</i> control. (Inherited from FlexListBase)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexListBase)
columnWidth	the width of the control's columns. (Inherited from FlexListBase)
currentState	the current view state of the component. (Inherited from FlexObject)
dataTipField	the name of the field in the data provider items to display as the data tip. (Inherited from FlexListBase)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)

Name	Description
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexListBase</i>)
<i>focusAlpha</i>	the alpha transparency value of the focus skin. The default value is 0.4. (Inherited from <i>FlexListBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>iconField</i>	the name of the field in the data provider object that determines what to display as the icon. (Inherited from <i>FlexListBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>labelField</i>	the name of the field in the data provider items to display as the label. (Inherited from <i>FlexListBase</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexListBase</i>)
<i>lockedColumnCount</i>	the index of the first column in the control that scrolls. (Inherited from <i>FlexListBase</i>)
<i>lockedRowCount</i>	the index of the first row in the control that scrolls. (Inherited from <i>FlexListBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexListBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexListBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)

Name	Description
<i>rollOverColor</i>	the color of the background of a renderer when the user rolls over it. The default value is 0xEEFEE6. (Inherited from <i>FlexListBase</i>)
<i>rowCount</i>	the number of rows to be displayed. (Inherited from <i>FlexListBase</i>)
<i>rowHeight</i>	the height of the rows in pixels. (Inherited from <i>FlexListBase</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectable</i>	whether the list shows selected items as selected. (Inherited from <i>FlexListBase</i>)
<i>selectedIndex</i>	the index in the data provider of the selected item. (Inherited from <i>FlexListBase</i>)
<i>selectedIndices</i>	an array of indices in the data provider of the selected items. (Inherited from <i>FlexListBase</i>)
<i>selectedItem</i>	a reference to the selected item in the data provider. (Inherited from <i>FlexListBase</i>)
<i>selectedItems</i>	an array of references to the selected items in the data provider. (Inherited from <i>FlexListBase</i>)
<i>selectionColor</i>	the color of the background of a renderer when the user selects it. The default value is 0x7FCEFF. (Inherited from <i>FlexListBase</i>)
<i>selectionDisabledColor</i>	the color of the background of a renderer when the component is disabled. The default value is 0xDDDDDD. (Inherited from <i>FlexListBase</i>)
<i>showDataTips</i>	whether dataTips are displayed for text in the rows. (Inherited from <i>FlexListBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textRollOverColor</i>	the text color of the label as the user moves the mouse pointer over the button. The default value is 0x2B333C. (Inherited from <i>FlexListBase</i>)
<i>textSelectedColor</i>	the color of the text of a renderer when the user selects it. The default value is 0x2B333C. (Inherited from <i>FlexListBase</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)

Name	Description
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
useRollOver	whether items are highlighted as the mouse rolls over them. (Inherited from FlexListBase)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
variableRowHeight	whether the individual rows can have different height. (Inherited from FlexListBase)
verticalAlign	the vertical alignment of a renderer in a row. (Inherited from FlexListBase)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
wordWrap	whether text in the row should be word wrapped. (Inherited from FlexListBase)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Deselect	Defines the value of the type property of the event object for an event that is dispatched when a previously selected item is deselected. (Inherited from FlexListBase)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Defines the value of the type property of the ListEvent object for an itemDoubleClick event, which indicates that the user double-clicked the mouse over a visual item in the control. (Inherited from FlexListBase)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexListBase)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexListBase)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexListBase)
Edit	Defines the value of the type property of the event object for a itemEditBegin event, which indicates that an item is ready to be edited.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexListBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexScrollBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
MultiSelect	Defines the value of the type property of the event object for an event that is dispatched when an item is selected as part of an action that selects multiple items. (Inherited from FlexListBase)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Scroll	Defines the value of the type property of the event object for a scroll event. (Inherited from FlexListBase)
ScrollToIndex	Ensures that the data provider item at the given index is visible. (Inherited from FlexListBase)
Select	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected. (Inherited from FlexListBase)
SelectIndex	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected. (Inherited from FlexListBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexListBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexListBase Class

Description

The `ListBase` class is the base class for controls that represent lists of items that can have one or more selected and can scroll through the items.

Items are supplied using the `dataProvider` property and displayed via item renderers.

In a model/view architecture, the `ListBase`-derived class represents the view, and the `dataProvider` object represents the model.

Inheritance Hierarchy

- [FlexScrollBase](#)
 - `FlexListBase`
 - [FlexAdvancedDataGrid](#)
 - [FlexDataGrid](#)

- [FlexList](#)

Syntax

```
'Declaration
Public Class FlexListBase _
Inherits FlexScrollBase
```

Properties

Name	Description
allowMultipleSelection	whether you can allow more than one item to be selected at the same time.
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
alternatingItemColors	the colors to use for the backgrounds of the items in the list.
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundDisabledColor	the color of text in the component if it is disabled. The default value is 0xAAB3B3.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnCount	the number of columns to be displayed in a <i>TileList</i> control or items in a <i>HorizontalList</i> control.
columnNames	a list containing the names of all columns in the data.
columnWidth	the width of the control's columns.
currentState	the current view state of the component. (Inherited from FlexObject)
dataTipField	the name of the field in the data provider items to display as the data tip.
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child.
focusAlpha	the alpha transparency value of the focus skin. The default value is 0.4.

Name	Description
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>iconField</i>	the name of the field in the data provider object that determines what to display as the icon.
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>labelField</i>	the name of the field in the data provider items to display as the label.
<i>lastVisibleRow</i>	the index of the last visible child.
<i>lockedColumnCount</i>	the index of the first column in the control that scrolls.
<i>lockedRowCount</i>	the index of the first row in the control that scrolls.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available.
<i>numRows</i>	the total number of rows of data available.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>rollOverColor</i>	the color of the background of a renderer when the user rolls over it. The default value is 0xEEFEE6.
<i>rowCount</i>	the number of rows to be displayed.
<i>rowHeight</i>	the height of the rows in pixels.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectable</i>	whether the list shows selected items as selected.
<i>selectedIndex</i>	the index in the data provider of the selected item.

Name	Description
<i>selectedIndices</i>	an array of indices in the data provider of the selected items.
<i>selectedItem</i>	a reference to the selected item in the data provider.
<i>selectedItems</i>	an array of references to the selected items in the data provider.
<i>selectionColor</i>	the color of the background of a renderer when the user selects it. The default value is 0x7FCEFF.
<i>selectionDisabledColor</i>	the color of the background of a renderer when the component is disabled. The default value is 0xDDDDDD.
<i>showDataTips</i>	whether dataTips are displayed for text in the rows.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textRollOverColor</i>	the text color of the label as the user moves the mouse pointer over the button. The default value is 0x2B333C.
<i>textSelectedColor</i>	the color of the text of a renderer when the user selects it. The default value is 0x2B333C.
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>useRollOver</i>	whether items are highlighted as the mouse rolls over them.
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>variableRowHeight</i>	whether the individual rows can have different height.
<i>verticalAlign</i>	the vertical alignment of a renderer in a row.
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>wordWrap</i>	whether text in the row should be word wrapped.
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Deselect	Defines the value of the type property of the event object for an event that is dispatched when a previously selected item is deselected.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Defines the value of the type property of the ListEvent object for an itemDoubleClick event, which indicates that the user double-clicked the mouse over a visual item in the control.
DragCancel	Defines the value of the type property of the event object for a dragComplete event.
DragDrop	Defines the value of the type property of the event object for a dragDrop event.
DragStart	Defines the value of the type property of the event object for a dragStart event.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexScrollBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)

Name	Description
<i>MultiSelect</i>	Defines the value of the type property of the event object for an event that is dispatched when an item is selected as part of an action that selects multiple items.
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Scroll</i>	Defines the value of the type property of the event object for a scroll event.
<i>ScrollToIndex</i>	Ensures that the data provider item at the given index is visible.
<i>Select</i>	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected.
<i>SelectIndex</i>	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected.
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>ToString</i>	Returns a string representation of the control. (Inherited from <i>FlexDisplayObject</i>)
<i>Type</i>	Dispatched when the user presses a key.
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexListLabel Class

Description

Defines the default item renderer for a control. By default, the item renderer draws the text associated with each item in the list, and an optional icon.

Inheritance Hierarchy

- [FlexDisplayObject](#)
 - FlexListLabel

Syntax

```
'Declaration  
Public Class FlexListLabel _  
Inherits FlexDisplayObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element.
automationIndex	a string containing the automation index for the element.
automationName	the name that can be used as an identifier for this object.
automationValue	the value that generally corresponds to the rendered appearance of the object and should be usable for correlating the identifier with the object as it appears visually within the application.
className	the name of this instance's class, such as "Button".
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)

Name	Description
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexLoader Class

Description

Loads and displays a specified SWF file. You typically use SWFLoader for loading one Flex application into a host Flex application.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexLoader
 - [FlexImage](#)

Syntax

```
'Declaration
Public Class FlexLoader _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)

Name	Description
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>source</i>	The URL, object, class or string name of a class to load as the content.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)

Name	Description
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexMenu Class

Description

Creates a pop-up menu of individually selectable choices, similar to the File or Edit menu found in most software applications. The popped up menu can have as many levels of submenus as needed.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexMenu

Syntax

```
'Declaration
Public Class FlexMenu _
Inherits FlexObject
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>alternatingItemColors</i>	the colors used for menu or submenu menu items in an alternating pattern.
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available.

Name	Description
<i>numRows</i>	the total number of rows of data available.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>rollOverColor</i>	the color of the menu item background when a user rolls the mouse over it. The default value is 0xB2E1FF.
<i>rowCount</i>	the number of rows to be displayed.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index in the data provider of the selected item.
<i>selectionColor</i>	the color of the menu item background when a menu item is selected. The default value is 0x7FCEFF.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textRollOverColor</i>	the color of the menu item text when a user rolls the mouse over the menu item. The default value is 0x2B333C.
<i>textSelectedColor</i>	the color of the menu item text when the menu item is selected. The default value is 0x2B333C.
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components.
Hide	Dispatched when a menu or submenu is dismissed.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Dispatched when a menu item is selected.

Name	Description
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Show	Dispatched when a menu or submenu opens.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexMenuBar Class

Description

Defines a horizontal, top-level menu bar that contains one or more menus. Clicking on a top-level menu item opens a pop-up submenu that is an instance of the Menu control.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexMenuBar

Syntax

```
'Declaration
Public Class FlexMenuBar _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)

Name	Description
<i>backgroundColor</i>	the background color of the component.
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>color</i>	the color of text in the component, including the component label. The default value is 0x0B333C.
<i>columnNames</i>	a list containing the names of all columns in the data.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>disabledColor</i>	the color of text in the component if it is disabled. The default value is 0xAAB3B3.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillColors</i>	the colors used to tint the background of the control.
<i>firstVisibleRow</i>	the index of the first visible child.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use.
<i>fontSize</i>	the height of the text, in pixels.
<i>fontStyle</i>	whether the text is italic font.
<i>fontWeight</i>	whether the text is boldface.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of children of this object.

Name	Description
<i>numColumns</i>	the total number of columns in the data available.
<i>numRows</i>	the total number of rows of data available.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>rollOverColor</i>	the color of the menu item background when a user rolls the mouse over it.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index in the MenuBar control of the currently open Menu or the last opened Menu if none are currently open.
<i>selectionColor</i>	the color of the menu item background when a menu item is selected.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textAlign</i>	the alignment of text within a container.
<i>textDecoration</i>	whether the text is underlined.
<i>textIndent</i>	the offset of the first line of text from the left side of the container, in pixels. The default value is 0.
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)

Name	Description
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
<i>GetValues</i>	Returns a matrix containing the automation values of all parts of the components.
<i>Hide</i>	Dispatched when a menu or submenu is dismissed.
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Show	Dispatched when a menu or submenu opens.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexNavigationBar Class

Description

The superclass for navigator controls, such as the LinkBar and TabBar controls. This class cannot be instantiated directly.

Inheritance Hierarchy

- [FlexBox](#)
 - FlexNavigationBar
 - [FlexButtonBar](#)
 - [FlexLinkBar](#)

Syntax

```
'Declaration
Public Class FlexNavigationBar _
Inherits FlexBox
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)

Name	Description
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>backgroundAlpha</i>	the alpha level of the color defined by the <i>backgroundColor</i> property, of the image or SWF file defined by the <i>backgroundImage</i> style. Valid values range from 0.0 to 1.0. (Inherited from <i>FlexContainer</i>)
<i>backgroundColor</i>	the background color of a component. (Inherited from <i>FlexContainer</i>)
<i>backgroundDisabledColor</i>	the background color of the component when it is disabled. The global default value is undefined. (Inherited from <i>FlexContainer</i>)
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class. (Inherited from <i>FlexContainer</i>)
<i>backgroundSize</i>	the percentage to change the image size to for the specified <i>backgroundImage</i> . (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>direction</i>	whether to use vertical (default) or horizontal layout. (Inherited from <i>FlexBox</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)

Name	Description
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAlign</i>	the horizontal alignment of children in the container. (Inherited from <i>FlexBox</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)

Name	Description
selectedIndex	the index of the active navigation item, where the first item is at an index of 0. The default value is -1.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAlign	the vertical alignment of children in the container. (Inherited from FlexBox)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
Change	Defines the value of the type property of the event object for an itemClick event.
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)

Name	Description
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)

Name	Description
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexNumericStepper Class

Description

Lets the user select a number from an ordered set.

The NumericStepper control consists of a single-line input text field and a pair of arrow buttons for stepping through the possible values. The Up Arrow and Down Arrow keys also cycle through the values.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexNumericStepper

Syntax

```
'Declaration
Public Class FlexNumericStepper _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
color	the color of text in the component, including the component label. The default value is 0x0B333C.

Name	Description
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>disabledColor</i>	the color of text in the component if it is disabled. The default value is 0xAAB3B3.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillColors</i>	the colors used to tint the background of the control. Pass the same color for both values for a flat-looking control. The default value is [0xFFFFFFFF, 0CCCCCC].
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use.
<i>fontSize</i>	the height of the text, in pixels.
<i>fontStyle</i>	whether the text is italic font.
<i>fontWeight</i>	whether the text is boldface. Recognized values are normal and bold.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>maximum</i>	the maximum value of the NumericStepper.
<i>minimum</i>	the minimum value of the NumericStepper.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>nextValue</i>	the value that is one step larger than the current value property and not greater than the maximum property value.
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)

Name	Description
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>previousValue</i>	the value that is one step smaller than the current value property and not smaller than the maximum property value.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>stepSize</i>	the non-zero unit of change between values.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textAlign</i>	the alignment of text within a container. Possible values are: left, right, or center.
<i>textDecoration</i>	whether the text is underlined. Possible values are <i>none</i> and <i>underline</i> . The default value is <i>none</i> .
<i>textIndent</i>	the offset of first line of text from the left side of the container, in pixels. The default value is 0.
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>value</i>	the value displayed in the text area of the NumericStepper control.
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Change</i>	Dispatched when the value of the NumericStepper control changes as a result of user interaction.
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Input	Creates an Event object that contains information about text events. Event objects are passed as parameters to event listeners.
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SelectText	Lets you track selection within a text field.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexObject Class

Description

The base class for all visual components, both interactive and noninteractive.

Inheritance Hierarchy

- [FlexDisplayObject](#)
 - FlexObject
 - [FlexAxisRenderer](#)
 - [FlexButton](#)
 - [FlexChart](#)
 - [FlexChartLegend](#)
 - [FlexChartSeries](#)
 - [FlexComboBase](#)
 - [FlexContainer](#)
 - [FlexDateChooser](#)
 - [FlexLabel](#)
 - [FlexLoader](#)
 - [FlexMenu](#)
 - [FlexMenuBar](#)
 - [FlexNumericStepper](#)
 - [FlexProgressBar](#)
 - [FlexRule](#)
 - [FlexScrollBar](#)

- [FlexScrollBase](#)
- [FlexSlider](#)
- [FlexVideoDisplay](#)
- [SparkDataGridLabel](#)
- [SparkGroupBase](#)
- [SparkObject](#)
- [SparkPopUpAnchor](#)
- [SparkTextBase](#)
- [SparkVideoDisplay](#)

Syntax

```
'Declaration
Public Class FlexObject _
Inherits FlexDisplayObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element.
automationIndex	a string containing the automation index for the element.
automationName	the name that can be used as an identifier for this object.
className	the name of this instance's class, such as <i>Button</i> .
currentState	the current view state of the component.
enabled	whether the component can accept user interaction.
errorColor	the color of the component highlight when validation fails.
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails.
focusEnabled	whether the component can receive focus when tabbed to.
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component.
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has.
percentHeight	the number that specifies the height of a component as a percentage of its parent's size.

Name	Description
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN.
scaleX	the number that specifies the horizontal scaling factor.
scaleY	the number that specifies the vertical scaling factor.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component.
toolTip	the text to display in the ToolTip. The default value is Nothing.
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead.
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
ChangeFocus	Changes the current focus.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexOLAPDataGrid Class

Description

Expands on the functionality of the AdvancedDataGrid control to add support for the display of the results of OLAP queries.

Like all Flex data grid controls, the OLAPDataGrid control is designed to display data in a two-dimensional representation of rows and columns.

Inheritance Hierarchy

- [FlexAdvancedDataGrid](#)
 - FlexOLAPDataGrid

Syntax

```
'Declaration  
Public Class FlexOLAPDataGrid _  
Inherits FlexAdvancedDataGrid
```

Properties

Name	Description
allowDragSelection	whether drag-selection is enabled. Drag-Selection is the ability to select an item by dragging into it as opposed to normal selection where you can't have the mouse button down when you mouse over the item you want to select. (Inherited from FlexAdvancedDataGrid)
allowMultipleSelection	whether you can allow more than one item to be selected at the same time. (Inherited from FlexListBase)
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
alternatingItemColors	the colors to use for the backgrounds of the items in the list. (Inherited from FlexListBase)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundDisabledColor	the color of text in the component if it is disabled. The default value is 0xAAB3B3. (Inherited from FlexListBase)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnCount	the number of columns to be displayed in a <i>TileList</i> control or items in a <i>HorizontalList</i> control. (Inherited from FlexListBase)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexListBase)
columnWidth	the width of the control's columns. (Inherited from FlexListBase)
currentState	the current view state of the component. (Inherited from FlexObject)
dataTipField	the name of the field in the data provider items to display as the data tip. (Inherited from FlexListBase)
defaultCellString	the String displayed in a cell when the data for that cell returned by the <i>IOLAPResult</i> instance is <i>Nothing</i> or <i>NaN</i> .

Name	Description
<i>depthColors</i>	the array of colors used for the rows of each level of the navigation tree of the AdvancedDataGrid control, in descending order. The default value is undefined. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>disabledColor</i>	the color of text in the component if it is disabled. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>displayItemsExpanded</i>	whether the navigation tree is expanded to show all items. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>editable</i>	whether the user can edit items in the data provider. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexListBase</i>)
<i>focusAlpha</i>	the alpha transparency value of the focus skin. The default value is 0.4. (Inherited from <i>FlexListBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>headerColors</i>	an array of two colors used to draw the header background gradient. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>headerHeight</i>	the height of the header cell of the column, in pixels. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalGridLineColor</i>	the color of the horizontal grid lines. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>iconField</i>	the name of the field in the data provider object that determines what to display as the icon. (Inherited from <i>FlexListBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>indentation</i>	the indentation for each node of the navigation tree, in pixels. The default value is 17. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>labelField</i>	the name of the field in the data provider items to display as the label. (Inherited from <i>FlexListBase</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexListBase</i>)
<i>lockedColumnCount</i>	the index of the first column in the control that scrolls. (Inherited from <i>FlexListBase</i>)

Name	Description
<i>lockedRowCount</i>	the index of the first row in the control that scrolls. (Inherited from <i>FlexListBase</i>)
<i>minColumnWidth</i>	the minimum width of the columns, in pixels. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexListBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexListBase</i>)
<i>openDuration</i>	the length of an open or close transition for the navigation tree, in milliseconds. The default value is 250. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>paddingBottom</i>	the number of pixels between the bottom of the row and the bottom of the renderer in the row. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>paddingTop</i>	the number of pixels between the top of the row and the top of the renderer in the row. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizableColumns</i>	whether the user can change the size of the columns. (Inherited from <i>FlexAdvancedDataGrid</i>)

Name	Description
<i>rollOverColor</i>	the color of the background of a renderer when the user rolls over it. The default value is 0xEEFEE6. (Inherited from <i>FlexListBase</i>)
<i>rowCount</i>	the number of rows to be displayed. (Inherited from <i>FlexListBase</i>)
<i>rowHeight</i>	the height of the rows in pixels. (Inherited from <i>FlexListBase</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectable</i>	whether the list shows selected items as selected. (Inherited from <i>FlexListBase</i>)
<i>selectedCells</i>	an Array of cell locations as row and column indices. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>selectedIndex</i>	the index in the data provider of the selected item. (Inherited from <i>FlexListBase</i>)
<i>selectedIndices</i>	an array of indices in the data provider of the selected items. (Inherited from <i>FlexListBase</i>)
<i>selectedItem</i>	a reference to the selected item in the data provider. (Inherited from <i>FlexListBase</i>)
<i>selectedItems</i>	an array of references to the selected items in the data provider. (Inherited from <i>FlexListBase</i>)
<i>selectionColor</i>	the color of the background of a renderer when the user selects it. The default value is 0x7FCEFF. (Inherited from <i>FlexListBase</i>)
<i>selectionDisabledColor</i>	the color of the background of a renderer when the component is disabled. The default value is 0xDDDDDD. (Inherited from <i>FlexListBase</i>)
<i>selectionMode</i>	the selection mode of the control. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>showDataTips</i>	whether dataTips are displayed for text in the rows. (Inherited from <i>FlexListBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textRollOverColor</i>	the text color of the label as the user moves the mouse pointer over the button. The default value is 0x2B333C. (Inherited from <i>FlexListBase</i>)

Name	Description
<i>textSelectedColor</i>	the color of the text of a renderer when the user selects it. The default value is 0x2B333C. (Inherited from <i>FlexListBase</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>useRollOver</i>	whether items are highlighted as the mouse rolls over them. (Inherited from <i>FlexListBase</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>variableRowHeight</i>	whether the individual rows can have different height. (Inherited from <i>FlexListBase</i>)
<i>verticalAlign</i>	the vertical alignment of a renderer in a row. (Inherited from <i>FlexListBase</i>)
<i>verticalGridLineColor</i>	the color of the vertical grid lines. The default value is 0x666666. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>wordWrap</i>	whether text in the row should be word wrapped. (Inherited from <i>FlexListBase</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)

Name	Description
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes or collapses a AdvancedDataGrid branch. (Inherited from FlexAdvancedDataGrid)
ColumnGroupedADGHeaderShift	Dispatched when the user releases the mouse button on a column header after having dragged the column to a new location resulting in shifting the column to a new index. (Inherited from FlexAdvancedDataGrid)
ColumnStretch	Dispatched when a user changes the width of a column, indicating that the amount of data displayed in that column may have changed. (Inherited from FlexAdvancedDataGrid)
Deselect	Defines the value of the type property of the event object for an event that is dispatched when a previously selected item is deselected. (Inherited from FlexListBase)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Defines the value of the type property of the ListEvent object for an itemDoubleClick event, which indicates that the user double-clicked the mouse over a visual item in the control. (Inherited from FlexListBase)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexListBase)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexListBase)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexListBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCellData	Returns data for a cell in the grid. (Inherited from FlexAdvancedDataGrid)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetGroupedItemChildrenCount	Returns the number of children within the first item of a group. (Inherited from FlexAdvancedDataGrid)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetRowData	Returns data for the row of the item in the data provider. (Inherited from FlexAdvancedDataGrid)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexListBase)
HeaderClick	Dispatched when the user releases the mouse button on a column header to request the control to sort the grid contents based on the contents of the column. (Inherited from FlexAdvancedDataGrid)
HeaderShift	Dispatched when the user releases the mouse button on a column header after having dragged the column to a new location resulting in shifting the column to a new index. (Inherited from FlexAdvancedDataGrid)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsGroupedItem</i>	Returns a value that specifies whether the item is a member of a group. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MouseScroll</i>	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from <i>FlexScrollBase</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>MultiSelect</i>	Defines the value of the type property of the event object for an event that is dispatched when an item is selected as part of an action that selects multiple items. (Inherited from <i>FlexListBase</i>)
<i>Open</i>	Opens or expands an AdvancedDataGrid branch. (Inherited from <i>FlexAdvancedDataGrid</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Scroll</i>	Defines the value of the type property of the event object for a scroll event. (Inherited from <i>FlexListBase</i>)
<i>ScrollToIndex</i>	Ensures that the data provider item at the given index is visible. (Inherited from <i>FlexListBase</i>)
<i>Select</i>	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected. (Inherited from <i>FlexListBase</i>)
<i>SelectIndex</i>	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected. (Inherited from <i>FlexListBase</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexListBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexPanel Class

Description

Consists of a title bar, a caption, a border, and a content area for its children.

Typically, you use Panel containers to wrap top-level application modules. For example, you can include a shopping cart in a Panel container.

Inheritance Hierarchy

- [FlexContainer](#)
 - FlexPanel
 - [FlexAlert](#)
 - [FlexTitleWindow](#)

Syntax

```
'Declaration
Public Class FlexPanel _
Inherits FlexContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)

Name	Description
<i>backgroundColor</i>	the background color of a component. (Inherited from <i>FlexContainer</i>)
<i>backgroundDisabledColor</i>	the background color of the component when it is disabled. The global default value is undefined. (Inherited from <i>FlexContainer</i>)
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class. (Inherited from <i>FlexContainer</i>)
<i>backgroundSize</i>	the percentage to change the image size to for the specified backgroundImage. (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dropShadowEnabled</i>	whether the Panel container's drop shadow is visible. The default value is true.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)

Name	Description
<i>footerColors</i>	the array of two colors used to draw the footer (area for the ControlBar container) background.
<i>headerColors</i>	the array of two colors used to draw the header.
<i>headerHeight</i>	the height of the header. The default value is based on the style of the title text.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>status</i>	the text in the status area of the title bar. The default value is "".
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
title	the title or caption displayed in the title bar. The default value is "".
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MouseScroll</i>	Dispatched when a mouse wheel is spun over an <i>InteractiveObject</i> instance. (Inherited from <i>FlexContainer</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Scroll</i>	Dispatched when the user manually scrolls the container. (Inherited from <i>FlexContainer</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexPieChart Class

Description

Represents a data series as a standard pie chart.

The data for the data provider determines the size of each wedge in the pie chart relative to the other wedges. You can use the `PieSeries` class to create standard pie charts, doughnut charts, or stacked pie charts.

Inheritance Hierarchy

- [FlexChart](#)
 - FlexPieChart

Syntax

```
'Declaration
Public Class FlexPieChart _
Inherits FlexChart
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipContent	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered. (Inherited from FlexChart)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChart)
currentState	the current view state of the component. (Inherited from FlexObject)
dataTipMode	how Flex displays <code>DataTip</code> controls for the chart. (Inherited from FlexChart)

Name	Description
<i>description</i>	a short description of the data in the chart. (Inherited from <i>FlexChart</i>)
<i>dragEnabled</i>	whether you can drag items out of this chart and drop them on other controls. (Inherited from <i>FlexChart</i>)
<i>dropEnabled</i>	whether dragged items can be dropped onto the chart. (Inherited from <i>FlexChart</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChart</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexChart</i>)
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11. (Inherited from <i>FlexChart</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>innerRadius</i>	the size of the hole in the center of the pie chart.
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChart</i>)
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show. (Inherited from <i>FlexChart</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>mouseSensitivity</i>	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart. (Inherited from <i>FlexChart</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChart</i>)

Name	Description
numRows	the total number of rows of data available. (Inherited from FlexChart)
paddingBottom	the number of pixels between the chart's bottom border and its content area. (Inherited from FlexChart)
paddingLeft	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from FlexChart)
paddingRight	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from FlexChart)
paddingTop	the number of pixels between the chart's top border and its content area. (Inherited from FlexChart)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedChartItem	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item. (Inherited from FlexChart)
selectedChartItems	an array of all the selected ChartItem objects in the chart. (Inherited from FlexChart)
selectionMode	whether ChartItem objects can be selected. (Inherited from FlexChart)
showDataTips	whether DataTip controls for the chart show. (Inherited from FlexChart)
showDataTipTargets	whether to show targets over the datapoints when showDataTips is set to true. (Inherited from FlexChart)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
textAlign	the alignment of text within a container.

Name	Description
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexChart)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexChart)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexChart)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChart)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header. (Inherited from FlexChart)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexPieSeries Class

Description

Defines the data series for a PieChart control.

Inheritance Hierarchy

- [FlexChartSeries](#)
 - FlexPieSeries

Syntax

```
'Declaration  
Public Class FlexPieSeries _  
Inherits FlexChartSeries
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
calloutGap	how much space, in pixels, to insert between the edge of the pie and the labels when rendering callouts. The default value is 10.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChartSeries)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
explodeRadius	a number from 0 to 1, specifying how far all wedges of the pie series should be exploded from the center of the chart as a percentage of the total radius.
field	the field of the data provider that determines the data for each wedge of the PieChart control. The default value is Nothing.
firstVisibleRow	the index of the first visible child. (Inherited from FlexChartSeries)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
fontSize	the height of the text, in pixels.

Name	Description
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>innerRadius</i>	a number from 0 to 1 specifying the distance from the center of the series to the inner edge of the rendered wedges, as a percentage of the total radius assigned to the series.
<i>insideLabelSizeLimit</i>	the font size threshold, in points, below which inside labels are considered illegible.
<i>labelPosition</i>	how to render value labels.
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from FlexChartSeries)
<i>maxLabelRadius</i>	the maximum amount of the PieSeries's radius that can be allocated to labels.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
<i>nameField</i>	the field of the data provider that determines the name of each wedge of the PieChart control. The default value is Nothing.
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from FlexChartSeries)
<i>numRows</i>	the total number of rows of data available. (Inherited from FlexChartSeries)
<i>outerRadius</i>	the percentage of the total space available to the PieSeries to use when rendering the contents of the series.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from FlexObject)

Name	Description
selectedIndex	the index of the selected item in the data provider of the series. (Inherited from FlexChartSeries)
selectedIndices	an array of indexes of the selected items in the data provider of the series. (Inherited from FlexChartSeries)
selectedItem	the chart item that is selected in the series. (Inherited from FlexChartSeries)
selectedItems	an array of chart items that are selected in the series. (Inherited from FlexChartSeries)
startAngle	the starting angle for the first slice of the PieChart control.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TextObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The

Name	Description
	<p>captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this <code>TestObject</code> . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChartSeries)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)

Name	Description
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter,

Name	Description
WaitForScreenshotStable	the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexPlotChart Class

Description

Represents data with two values for each data point. One value determines the position of the data point along the horizontal axis, and one value determines its position along the vertical axis.

Inheritance Hierarchy

- [FlexCartesianChart](#)
 - FlexPlotChart

Syntax

```
'Declaration
Public Class FlexPlotChart _
Inherits FlexCartesianChart
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipContent	whether Flex clips the chart to the area bounded by the axes. Set to false to clip the chart. Set to true to avoid clipping when the data is rendered. (Inherited from FlexChart)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexChart)

Name	Description
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dataTipMode</i>	how Flex displays DataTip controls for the chart. (Inherited from <i>FlexChart</i>)
<i>description</i>	a short description of the data in the chart. (Inherited from <i>FlexChart</i>)
<i>dragEnabled</i>	whether you can drag items out of this chart and drop them on other controls. (Inherited from <i>FlexChart</i>)
<i>dropEnabled</i>	whether dragged items can be dropped onto the chart. (Inherited from <i>FlexChart</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChart</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexChart</i>)
<i>fontSize</i>	the height of the text, in pixels. The default value is 10 for all controls except the ColorPicker control. For the ColorPicker control, the default value is 11. (Inherited from <i>FlexChart</i>)
<i>gutterBottom</i>	The size of the region, in pixels, between the bottom of the chart data area and the bottom of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterLeft</i>	The size of the region, in pixels, between the left of the chart data area and the left of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterRight</i>	The size of the region, in pixels, between the right side of the chart data area and the outside of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>gutterTop</i>	The size of the region, in pixels, between the top of the chart data area and the top of the chart control. (Inherited from <i>FlexCartesianChart</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAxisRatio</i>	the height limit of the horizontal axis. (Inherited from <i>FlexCartesianChart</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChart</i>)

Name	Description
<i>maximumDataTipCount</i>	the maximum number of datatips a chart will show. (Inherited from <i>FlexChart</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>mouseSensitivity</i>	the distance, in pixels, that Flex considers a data point to be under the mouse pointer when the pointer moves around a chart. (Inherited from <i>FlexChart</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChart</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChart</i>)
<i>paddingBottom</i>	the number of pixels between the chart's bottom border and its content area. (Inherited from <i>FlexChart</i>)
<i>paddingLeft</i>	the number of pixels between the control's left border and the left edge of its content area. The default value is 2. (Inherited from <i>FlexChart</i>)
<i>paddingRight</i>	the number of pixels between the control's right border and the right edge of its content area. The default value is 0. (Inherited from <i>FlexChart</i>)
<i>paddingTop</i>	the number of pixels between the chart's top border and its content area. (Inherited from <i>FlexChart</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedChartItem</i>	the selected ChartItem in the chart. If multiple items are selected, this property specifies the most recently selected item. (Inherited from <i>FlexChart</i>)

Name	Description
<i>selectedChartItems</i>	an array of all the selected <i>ChartItem</i> objects in the chart. (Inherited from <i>FlexChart</i>)
<i>selectionMode</i>	whether <i>ChartItem</i> objects can be selected. (Inherited from <i>FlexChart</i>)
<i>showDataTips</i>	whether <i>DataTip</i> controls for the chart show. (Inherited from <i>FlexChart</i>)
<i>showDataTipTargets</i>	whether to show targets over the datapoints when <i>showDataTips</i> is set to true. (Inherited from <i>FlexChart</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textAlign</i>	the alignment of text within a container. (Inherited from <i>FlexCartesianChart</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the <i>ToolTip</i> . The default value is <i>Nothing</i> . (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If <i>useHandCursor</i> is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>verticalAxisRatio</i>	the width limit of the vertical axis. (Inherited from <i>FlexCartesianChart</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the <i>DisplayObject</i> instance relative to the local coordinates of the parent <i>DisplayObjectContainer</i> . (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the <i>DisplayObject</i> instance relative to the local coordinates of the parent <i>DisplayObjectContainer</i> . (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexChart)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexChart)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexChart)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChart)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Defines the value of the type property of the event object for a headerRelease event, which indicates that the user pressed and released the mouse on a column header. (Inherited from FlexChart)

Name	Description
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by

Name	Description
WaitForPropertyNotEquals	the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexPlotSeries Class

Description

Defines a data series for a PlotChart control.

Inheritance Hierarchy

- [FlexChartSeries](#)
 - FlexPlotSeries

Syntax

```
'Declaration
Public Class FlexPlotSeries _
Inherits FlexChartSeries
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)

Name	Description
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexChartSeries</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexChartSeries</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexChartSeries</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexChartSeries</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexChartSeries</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>radius</i>	the radius style

Name	Description
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index of the selected item in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedIndices</i>	an array of indexes of the selected items in the data provider of the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedItem</i>	the chart item that is selected in the series. (Inherited from <i>FlexChartSeries</i>)
<i>selectedItems</i>	an array of chart items that are selected in the series. (Inherited from <i>FlexChartSeries</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>xField</i>	the field of the data provider that determines the x-axis location of each data point.
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>yField</i>	the field of the data provider that determines the y-axis location of each data point.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)

Name	Description
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexChartSeries)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SeriesClick	Indicates that the user clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesDoubleClick	Indicates that the user double-clicked the mouse button over a chart item representing data in the chart. (Inherited from FlexChartSeries)
SeriesItemRollOver	Indicates that the user rolled the mouse pointer over a chart item representing data in the chart. (Inherited from FlexChartSeries)

Name	Description
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by

Name	Description
WaitForPropertyNotEquals	the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexPopUpButton Class

Description

The PopUpButton control adds a flexible pop-up control interface to a Button control. It contains a main button and a secondary button, called the pop-up button, which pops up any UIComponent object when a user clicks the pop-up button.

A PopUpButton control can have a text label, an icon, or both on its face. When a user clicks the main part of the PopUpButton control, it dispatches a click event.

One common use for the PopUpButton control is to have the pop-up button open a List control or a Menu control that changes the function and label of the main button.

Inheritance Hierarchy

- [FlexButton](#)
 - FlexPopUpButton

Syntax

```
'Declaration
Public Class FlexPopUpButton _
Inherits FlexButton
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)

Name	Description
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>color</i>	the color of text in the component, including the component label. The default value is 0x0B333C. (Inherited from <i>FlexButton</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>disabledColor</i>	the color of text in the component if it is disabled. The default value is 0xAAB3B3. (Inherited from <i>FlexButton</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alphas used for the background fill of controls. (Inherited from <i>FlexButton</i>)
<i>fillColors</i>	the colors used to tint the background of the control. (Inherited from <i>FlexButton</i>)
<i>focusAlpha</i>	the alpha transparency value of the focus skin. The default value is 0.4. (Inherited from <i>FlexButton</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexButton</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexButton</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexButton</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexButton</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text to appear on the Button control. (Inherited from <i>FlexButton</i>)
<i>labelPlacement</i>	the orientation of the label in relation to a specified icon. (Inherited from <i>FlexButton</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list

Name	Description
	are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selected	whether a toggle button is toggled on (true) or off (false). (Inherited from FlexButton)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
textAlign	the alignment of text within a container. (Inherited from FlexButton)
textDecoration	whether the text is underlined. (Inherited from FlexButton)
textIndent	the offset of the first line of text from the left side of the container, in pixels. The default value is 0. (Inherited from FlexButton)
textRollOverColor	the text color of the label as the user moves the mouse pointer over the button. The default value is 0x2B333C. (Inherited from FlexButton)
textSelectedColor	the text color of the label as the user presses it. The default value is 0x000000. (Inherited from FlexButton)
themeColor	the theme color of a component. (Inherited from FlexObject)
toggle	whether a Button is in a toggle state or not. (Inherited from FlexButton)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)

Name	Description
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Dispatched when the specified UIComponent closes.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
Open	Dispatched when the specified UIComponent opens.

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexButton)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException

Name	Description
	<p>is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexProgressBar Class

Description

The `ProgressBar` control provides a visual representation of the progress of a task over time. There are two types of `ProgressBar` controls: determinate and indeterminate.

A determinate `ProgressBar` control is a linear representation of the progress of a task over time. You use a determinate `ProgressBar` when the scope of the task is known. It displays when the user has to wait for an extended amount of time.

An indeterminate `ProgressBar` control represents time-based processes for which the scope is not yet known. As soon as you can determine the scope, you should use a determinate `ProgressBar` control.

Inheritance Hierarchy

- [FlexObject](#)
 - `FlexProgressBar`

Syntax

```
'Declaration  
Public Class FlexProgressBar _  
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
barColor	the color of a ProgressBar.
borderColor	the black section of a three-dimensional border, or the color section of a two-dimensional border. The default value is 0xB7BABC.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
color	the color of text in the component, including the component label. The default value is 0x0B333C.
conversion	the number used to convert incoming current bytes loaded value and the total bytes loaded values.
currentState	the current view state of the component. (Inherited from FlexObject)
direction	the direction in which the fill of the ProgressBar expands toward completion. Valid values in MXML are "right" and "left".
disabledColor	the color of text in the component if it is disabled. The default value is 0xAAB3B3.
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
fillColors	the colors used to tint the background fill of the component.
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
fontFamily	the name of the font to use.
fontSize	the height of the text, in pixels.

Name	Description
<i>fontStyle</i>	whether the text is italic font.
<i>fontWeight</i>	whether the text is boldface.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>indeterminate</i>	whether the ProgressBar control has a determinate or indeterminate appearance.
<i>label</i>	the text that accompanies the progress bar.
<i>labelPlacement</i>	the placement of the label.
<i>labelWidth</i>	the width of the label in pixels.
<i>maximum</i>	the largest progress value for the ProgressBar. You can only use this property in manual mode.
<i>minimum</i>	the smallest progress value for the ProgressBar. This property is set by the developer only in manual mode.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentComplete</i>	the percentage of process that is completed. The range is 0 to 100.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
textAlign	the alignment of text within a container.
textDecoration	whether the text is underlined.
textIndent	the offset of first line of text from the left side of the container, in pixels. The default value is 0.
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
value	the amount of progress that has been made between the minimum and maximum values.
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForProperty	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexRadioButton Class

Description

The RadioButton control lets the user make a single choice within a set of mutually exclusive choices.

Inheritance Hierarchy

- [FlexButton](#)
 - FlexRadioButton

Syntax

```
'Declaration  
Public Class FlexRadioButton _  
Inherits FlexButton
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
color	the color of text in the component, including the component label. The default value is 0x0B333C. (Inherited from FlexButton)
currentState	the current view state of the component. (Inherited from FlexObject)
disabledColor	the color of text in the component if it is disabled. The default value is 0xAAB3B3. (Inherited from FlexButton)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
fillAlphas	the alphas used for the background fill of controls. (Inherited from FlexButton)
fillColors	the colors used to tint the background of the control. (Inherited from FlexButton)
focusAlpha	the alpha transparency value of the focus skin. The default value is 0.4. (Inherited from FlexButton)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
fontFamily	the name of the font to use. (Inherited from FlexButton)
fontSize	the height of the text, in pixels. (Inherited from FlexButton)
fontStyle	whether the text is italic font. (Inherited from FlexButton)
fontWeight	whether the text is boldface. (Inherited from FlexButton)

Name	Description
<i>groupName</i>	the name of the group to which this RadioButton control belongs, or specifies the value of the id property of a RadioButtonGroup control if this RadioButton is part of a group defined by a RadioButtonGroup control.
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>label</i>	the text to appear on the Button control. (Inherited from FlexButton)
<i>labelPlacement</i>	the orientation of the label in relation to a specified icon. (Inherited from FlexButton)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
<i>selected</i>	whether a toggle button is toggled on (true) or off (false). (Inherited from FlexButton)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from FlexDisplayObject)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
<i>Text</i>	The text of the control. (Inherited from TestObject)
<i>textAlign</i>	the alignment of text within a container. (Inherited from FlexButton)

Name	Description
textDecoration	whether the text is underlined. (Inherited from FlexButton)
textIndent	the offset of the first line of text from the left side of the container, in pixels. The default value is 0. (Inherited from FlexButton)
textRollOverColor	the text color of the label as the user moves the mouse pointer over the button. The default value is 0x2B333C. (Inherited from FlexButton)
textSelectedColor	the text color of the label as the user presses it. The default value is 0x000000. (Inherited from FlexButton)
themeColor	the theme color of a component. (Inherited from FlexObject)
toggle	whether a Button is in a toggle state or not. (Inherited from FlexButton)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>ToString</i>	Returns a string representation of the control. (Inherited from <i>FlexDisplayObject</i>)
<i>Type</i>	Dispatched when the user presses a key. (Inherited from <i>FlexButton</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexRepeater Class

Description

Identifies the repeater control, which creates multiple instances of its subcomponents.

Inheritance Hierarchy

- [TestObject](#)
 - FlexRepeater

Syntax

```
'Declaration
Public Class FlexRepeater _
Inherits TestObject _
Implements IClickable, IKeyable
```

Properties

Name	Description
automationClassName	a string containing the automation class name for the element.
automationIndex	a string containing the automation index for the element.
automationName	the name that can be used as an identifier for this object.
className	a string containing the class name of the element as assigned by the control developer.
columnNames	a list containing the names of all columns in the data.
count	the number of times this Repeater should execute.
firstVisibleRow	the index of the first visible child.
id	a string containing the automation identifier (ID) for the element.
lastVisibleRow	the index of the last visible child.
numColumns	the total number of columns in the data available.
numRows	the total number of rows of data available.
recycleChildren	whether this Repeater should re-use previously created children, or create new ones.
startingIndex	the index into the dataProvider at which this Repeater starts creating children.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetValues	Returns a matrix containing the automation values of all parts of the components.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexRule Class

Description

Creates a single horizontal or vertical line. You typically use this control to create a dividing line within a container.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexRule

Syntax

```
'Declaration
Public Class FlexRule _
Inherits FlexObject
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>color</i>	the color of the line.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)

Name	Description
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>strokeWidth</i>	the thickness of the rule in pixels.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexScrollBar Class

Description

Lets you control the portion of data that is displayed when there is too much data to fit in a display area.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexScrollBar

Syntax

```
'Declaration  
Public Class FlexScrollBar _  
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
fillColors	the colors used to tint the background of the control.
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
lineScrollSize	the amount to scroll when an arrow button is pressed, in pixels. The default value is 1.
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all

Name	Description
	children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>pageScrollSize</i>	the amount to move the scroll thumb when the scroll bar track is pressed, in pixels. The default value is 0.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or <code>explicitWidth</code> properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>scrollPosition</i>	the number that represents the current scroll position.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>trackColors</i>	The colors of the track, as an array of two colors.
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If <code>useHandCursor</code> is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the ScrollBar control scrolls through user initiated action or programmatically.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 ToString 	Returns a string representation of the control. (Inherited from FlexDisplayObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An

Name	Description
WaitForPropertyNotEquals	exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexScrollBase Class

Description

The ScrollControlBase class is the base class for controls with scroll bars.

The user interacts with the scroll bar or the developer accesses methods and properties that alter the viewable area. The ScrollControlBase takes a single child object and positions and masks or sizes that object to display the viewable content. All items to be scrolled must be children of that content object.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexScrollBase
 - [FlexListBase](#)
 - [FlexTextArea](#)
 - [SparkAirHTML](#)

Syntax

```
'Declaration
Public Class FlexScrollBase _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)

Name	Description
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is

Name	Description
	passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance.
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexSlider Class

Description

Lets users select a value by moving a slider thumb between the end points of the slider track.

The current value of the slider is determined by the relative location of the thumb between the end points of the slider, corresponding to the slider's minimum and maximum values.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexSlider

Syntax

```
'Declaration  
Public Class FlexSlider _  
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
dataTipPrecision	the number of decimal places to use for the data tip text.
direction	the orientation of the slider control.
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
fillColors	the colors used to tint the background of the control.
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
labels	an array of strings used for the slider labels.
labelStyleName	the name of the style to use for the slider label. The default value is undefined.
liveDragging	whether live dragging is enabled for the slider.
maximum	the maximum allowed value on the slider. The default value is 10.
minimum	the minimum allowed value on the slider control. The default value is 0.

Name	Description
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>showDataTip</i>	whether the data tip shows during user interaction.
<i>snapInterval</i>	the increment value of the slider thumb as the user moves the thumb.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>thumbCount</i>	the number of thumbs allowed on the slider.
<i>tickColor</i>	the color of the tick marks.
<i>tickLength</i>	the length in pixels of the tick marks.
<i>tickThickness</i>	the thickness in pixels of the tick marks.
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>trackColors</i>	the colors of the track, as an array of two colors.
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is

Name	Description
	false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
value	the position of the thumb, and is a number between the minimum and maximum properties.
values	an array of values for each thumb when thumbCount is greater than 1.
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when the slider changes value due to mouse or keyboard interaction.
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexStandalonePlayer Class

Description

The Flex Standalone Player.

Inheritance Hierarchy

- [Window](#)
 - FlexStandalonePlayer

Syntax

```
'Declaration
Public Class FlexStandalonePlayer _
Inherits Window
```

Properties

Name	Description
Application	The name of the Application that this Window belongs to. (Inherited from Window)
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsActive</i>	Returns whether the window is set to active. (Inherited from <i>IMoveable</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>Maximize</i>	Maximizes the window. (Inherited from <i>IMoveable</i>)
<i>Minimize</i>	Reduces the window to an icon. (Inherited from <i>IMoveable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the window. (Inherited from <i>IMoveable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Restore</i>	Restores the window to its previous size. (Inherited from <i>IMoveable</i>)
<i>SetActive</i>	Makes the window active. (Inherited from <i>IMoveable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>Size</i>	Resizes the window. (Inherited from <i>IMoveable</i>)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexTabNavigator Class

Description

Extends the ViewStack container by including a TabBar container for navigating between its child containers.

Inheritance Hierarchy

- [FlexViewStack](#)
 - FlexTabNavigator

Syntax

```
'Declaration
Public Class FlexTabNavigator _
Inherits FlexViewStack
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)
backgroundDisabledColor	the background color of the component when it is disabled. The global default value is undefined. (Inherited from FlexContainer)

Name	Description
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class. (Inherited from <i>FlexContainer</i>)
<i>backgroundSize</i>	the percentage to change the image size to for the specified backgroundImage. (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAlign</i>	the horizontal positioning of tabs at the top of this TabNavigator container.
<i>horizontalGap</i>	the separation between tabs, in pixels. The default value is -1, so that the borders of adjacent tabs overlap.
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)

Name	Description
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedChild</i>	a reference to the currently visible child container. (Inherited from <i>FlexViewStack</i>)
<i>selectedIndex</i>	the zero-based index of the currently visible child container. (Inherited from <i>FlexViewStack</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabHeight</i>	the height of each tab, in pixels.
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>tabWidth</i>	the width of each tab, in pixels.
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Change</i>	Defines the value of the type property of the event object for an itemClick event.
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>DragCancel</i>	Defines the value of the type property of the event object for a dragComplete event. (Inherited from <i>FlexContainer</i>)

Name	Description
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexViewStack)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexTextArea Class

Description

A multiline text field with a border and optional scroll bars.

The TextArea control supports the HTML rendering capabilities of Flash Player and AIR.

Inheritance Hierarchy

- [FlexScrollBase](#)
 - FlexTextArea

Syntax

```
'Declaration
Public Class FlexTextArea _
Inherits FlexScrollBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
color	the color of text in the component, including the component label. The default value is 0x0B333C.
currentState	the current view state of the component. (Inherited from FlexObject)
disabledColor	the background color of a component.
displayAsPassword	whether this control is used for entering passwords.
editable	whether the user can edit the text in this control.
enabled	whether the component can accept user interaction. (Inherited from FlexObject)

Name	Description
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use.
<i>fontSize</i>	the height of the text, in pixels.
<i>fontStyle</i>	whether the text is italic font.
<i>fontWeight</i>	whether the text is boldface.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalScrollPolicy</i>	whether the horizontal scroll bar is always on (ScrollPolicy.ON), always off (ScrollPolicy.OFF), or turns on when needed (ScrollPolicy.AUTO).
<i>htmlText</i>	the text displayed by the TextInput control, including HTML markup that expresses the styles of that text.
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>length</i>	the number of characters of text displayed in the TextArea.
<i>maxChars</i>	the maximum number of characters that users can enter in the text field.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>restrict</i>	the set of characters that a user can enter into the control.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)

Name	Description
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectionBeginIndex	the zero-based character index value of the first character in the current selection.
selectionEndIndex	the zero-based index of the position after the last character in the current selection (equivalent to the one-based index of the last character).
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
text	the plain text that appears in the control.
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
wordWrap	whether text in the row should be word wrapped.
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The

Name	Description
	<p>captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this <code>TestObject</code> . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Input</i>	Dispatched when the user types, deletes, or pastes text into the control.
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MouseScroll</i>	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from <i>FlexScrollBase</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Scroll</i>	Defines the value of the type property of the event object for a scroll event.
<i>SelectText</i>	Lets you track selection within a text field.
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexTitleWindow Class

Description

Contains a title bar, a caption, a border, and a content area for its child.

Typically, you use TitleWindow containers to wrap self-contained application modules. For example, you can include a form in a TitleWindow container. When the user completes the form, you can close the TitleWindow container programmatically, or let the user close it by using the Close button.

Inheritance Hierarchy

- [FlexPanel](#)
 - FlexTitleWindow

Syntax

```
'Declaration
Public Class FlexTitleWindow _
Inherits FlexPanel
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)

Name	Description
<i>backgroundDisabledColor</i>	the background color of the component when it is disabled. The global default value is undefined. (Inherited from <i>FlexContainer</i>)
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class. (Inherited from <i>FlexContainer</i>)
<i>backgroundSize</i>	the percentage to change the image size to for the specified <i>backgroundImage</i> . (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dropShadowEnabled</i>	whether the Panel container's drop shadow is visible. The default value is true. (Inherited from <i>FlexPanel</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>footerColors</i>	the array of two colors used to draw the footer (area for the ControlBar container) background. (Inherited from <i>FlexPanel</i>)

Name	Description
<i>headerColors</i>	the array of two colors used to draw the header. (Inherited from <i>FlexPanel</i>)
<i>headerHeight</i>	the height of the header. The default value is based on the style of the title text. (Inherited from <i>FlexPanel</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>showCloseButton</i>	whether to display a Close button in the TitleWindow container. The default value is false.
<i>status</i>	the text in the status area of the title bar. The default value is "". (Inherited from <i>FlexPanel</i>)

Name	Description
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
title	the title or caption displayed in the title bar. The default value is "". (Inherited from FlexPanel)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexToggleButtonBar Class

Description

Defines a horizontal or vertical group of buttons that maintain their selected or deselected state.

Only one button in the ToggleButtonBar control can be in the selected state. This means that when a user selects a button in a ToggleButtonBar control, the button stays in the selected state until the user selects a different button.

Inheritance Hierarchy

- [FlexButtonBar](#)
 - FlexToggleButtonBar

Syntax

```
'Declaration
Public Class FlexToggleButtonBar _
Inherits FlexButtonBar
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)
backgroundDisabledColor	the background color of the component when it is disabled. The global default value is undefined. (Inherited from FlexContainer)
backgroundImage	the background image of a component. This can be an absolute or relative URL or class. (Inherited from FlexContainer)

Name	Description
<i>backgroundSize</i>	the percentage to change the image size to for the specified backgroundImage. (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>direction</i>	whether to use vertical (default) or horizontal layout. (Inherited from <i>FlexBox</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAlign</i>	the horizontal alignment of children in the container. (Inherited from <i>FlexBox</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)

Name	Description
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	the index of the active navigation item, where the first item is at an index of 0. The default value is -1. (Inherited from <i>FlexNavigationBar</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)

Name	Description
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalAlign	the vertical alignment of children in the container. (Inherited from FlexBox)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Defines the value of the type property of the event object for an itemClick event. (Inherited from FlexNavigationBar)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)

Name	Description
Type	Dispatched when the user presses a key. (Inherited from FlexNavigationBar)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexTree Class

Description

The Tree control lets a user view hierarchical data arranged as an expandable tree.

Each item in a tree can be a leaf or a branch. A leaf item is an end point in the tree. A branch item can contain leaf or branch items, or it can be empty.

By default, a leaf is represented by a text label next to a file icon. A branch is represented by a text label next to a folder icon, with a disclosure triangle that a user can open to expose children.

Inheritance Hierarchy

- [FlexList](#)
 - FlexTree

Syntax

```
'Declaration  
Public Class FlexTree _  
Inherits FlexList
```

Properties

Name	Description
allowMultipleSelection	whether you can allow more than one item to be selected at the same time. (Inherited from FlexListBase)
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
alternatingItemColors	the colors to use for the backgrounds of the items in the list. (Inherited from FlexListBase)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundDisabledColor	the color of text in the component if it is disabled. The default value is 0xAAB3B3. (Inherited from FlexListBase)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnCount	the number of columns to be displayed in a <i>TileList</i> control or items in a <i>HorizontalList</i> control. (Inherited from FlexListBase)
columnNames	a list containing the names of all columns in the data. (Inherited from FlexListBase)
columnWidth	the width of the control's columns. (Inherited from FlexListBase)

Name	Description
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dataTipField</i>	the name of the field in the data provider items to display as the data tip. (Inherited from <i>FlexListBase</i>)
<i>depthColors</i>	the array of colors used in the Tree control, in descending order. The default value is undefined.
<i>disabledColor</i>	the disabled color of a list item. The default value is 0xDDDDDD.
<i>editable</i>	whether the user can edit items in the data provider.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexListBase</i>)
<i>focusAlpha</i>	the alpha transparency value of the focus skin. The default value is 0.4. (Inherited from <i>FlexListBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>iconField</i>	the name of the field in the data provider object that determines what to display as the icon. (Inherited from <i>FlexListBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>indentation</i>	the indentation for each tree level, in pixels. The default value is 17.
<i>labelField</i>	the name of the field in the data provider items to display as the label. (Inherited from <i>FlexListBase</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexListBase</i>)
<i>lockedColumnCount</i>	the index of the first column in the control that scrolls. (Inherited from <i>FlexListBase</i>)
<i>lockedRowCount</i>	the index of the first row in the control that scrolls. (Inherited from <i>FlexListBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use

Name	Description
	flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>FlexListBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexListBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>rollOverColor</i>	the color of the background of a renderer when the user rolls over it. The default value is 0xEEFEE6. (Inherited from <i>FlexListBase</i>)
<i>rowCount</i>	the number of rows to be displayed. (Inherited from <i>FlexListBase</i>)
<i>rowHeight</i>	the height of the rows in pixels. (Inherited from <i>FlexListBase</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectable</i>	whether the list shows selected items as selected. (Inherited from <i>FlexListBase</i>)
<i>selectedIndex</i>	the index in the data provider of the selected item. (Inherited from <i>FlexListBase</i>)
<i>selectedIndices</i>	an array of indices in the data provider of the selected items. (Inherited from <i>FlexListBase</i>)
<i>selectedItem</i>	a reference to the selected item in the data provider. (Inherited from <i>FlexListBase</i>)
<i>selectedItems</i>	an array of references to the selected items in the data provider. (Inherited from <i>FlexListBase</i>)
<i>selectionColor</i>	the color of the background of a renderer when the user selects it. The default value is 0x7FCEFF. (Inherited from <i>FlexListBase</i>)
<i>selectionDisabledColor</i>	the color of the background of a renderer when the component is disabled. The default value is 0xDDDDDD. (Inherited from <i>FlexListBase</i>)
<i>showDataTips</i>	whether dataTips are displayed for text in the rows. (Inherited from <i>FlexListBase</i>)

Name	Description
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>textRollOverColor</i>	the text color of the label as the user moves the mouse pointer over the button. The default value is 0x2B333C. (Inherited from <i>FlexListBase</i>)
<i>textSelectedColor</i>	the color of the text of a renderer when the user selects it. The default value is 0x2B333C. (Inherited from <i>FlexListBase</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>useRollOver</i>	whether items are highlighted as the mouse rolls over them. (Inherited from <i>FlexListBase</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>variableRowHeight</i>	whether the individual rows can have different height. (Inherited from <i>FlexListBase</i>)
<i>verticalAlign</i>	the vertical alignment of a renderer in a row. (Inherited from <i>FlexListBase</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>wordWrap</i>	whether text in the row should be word wrapped. (Inherited from <i>FlexListBase</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes or collapses a tree branch.
Deselect	Defines the value of the type property of the event object for an event that is dispatched when a previously selected item is deselected. (Inherited from FlexListBase)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Defines the value of the type property of the ListEvent object for an itemDoubleClick event, which indicates that the user double-clicked the mouse over a visual item in the control. (Inherited from FlexListBase)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexListBase)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexListBase)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexListBase)
Edit	Defines the value of the type property of the event object for a itemEditBegin event, which indicates that an item is ready to be edited. (Inherited from FlexList)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from FlexListBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexScrollBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)

Name	Description
MultiSelect	Defines the value of the type property of the event object for an event that is dispatched when an item is selected as part of an action that selects multiple items. (Inherited from FlexListBase)
Open	Opens or expands a tree branch.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Defines the value of the type property of the event object for a scroll event. (Inherited from FlexListBase)
ScrollToIndex	Ensures that the data provider item at the given index is visible. (Inherited from FlexListBase)
Select	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected. (Inherited from FlexListBase)
SelectIndex	Defines the value of the type property of the event object for an event that is dispatched when a single item is selected. (Inherited from FlexListBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from FlexListBase)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexUIMovieClip Class

Description

Components created in Adobe Flash CS3 Professional for use in Flex are subclasses of the mx.flash.UIMovieClip class.

The UIMovieClip class implements the interfaces necessary for a Flash component to be used like a normal Flex component. Therefore, a subclass of UIMovieClip can be used as a child of a Flex container or as a skin, and it can respond to events, define view states and transitions, and work with effects in the same way as can any Flex component.

Inheritance Hierarchy

- [FlexDisplayObject](#)
 - FlexUIMovieClip
 - [FlexContainerMovieClip](#)

Syntax

```
'Declaration  
Public Class FlexUIMovieClip _  
Inherits FlexDisplayObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element.
automationIndex	a string containing the automation index for the element.
automationName	the name that can be used as an identifier for this object.
className	the name of this instance's class, such as "Button".
currentState	the current state of this component.
enabled	whether a movie clip is enabled.
errorColor	the color of the component highlight when validation fails.
errorString	the text that is displayed by a component's error tip when a component is monitored and validation fails.
focusEnabled	whether the component can receive focus when selected.
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component.
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any

Name	Description
	mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has.
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. Valid values are 0 to 100.
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Valid values are 0 to 100.
scaleX	the horizontal scale (percentage) of the object as applied from the registration point. The default registration point is (0,0). 1.0 equals 100% scale.
scaleY	the vertical scale (percentage) of an object as applied from the registration point of the object. The default registration point is (0,0). 1.0 is 100% scale.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component.
toolTip	the text to display in the ToolTip. The default value is Nothing.
useHandCursor	whether the pointing hand (hand cursor) appears when the mouse rolls over a sprite in which the <code>buttonMode</code> property is set to true.
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Defines the value of the type property of a keyFocusChange event object.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Called by the FocusManager when the component receives focus. The component may in turn set focus to an internal component.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexVideoDisplay Class

Description

Lets you play an FLV file in a Flex application. It supports progressive download over HTTP, streaming from the Flash Media Server, and streaming from a Camera object.

Inheritance Hierarchy

- [FlexObject](#)
 - FlexVideoDisplay

Syntax

```
'Declaration
Public Class FlexVideoDisplay _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)

Name	Description
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
source	the relative path and filename of the FLV file to stream.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
totalTime	the total length of the media, in seconds.
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>volume</i>	the volume level, specified as an value between 0 and 1. The default value is 0.75.
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from <i>FlexDisplayObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexViewStack Class

Description

Consists of a collection of child containers stacked on top of each other, where only one child at a time is visible. When a different child container is selected, it seems to replace the old one because it appears in the same location. However, the old child container still exists; it is just invisible.

Inheritance Hierarchy

- [FlexContainer](#)
 - FlexViewStack
 - [FlexTabNavigator](#)

Syntax

```
'Declaration
Public Class FlexViewStack _
Inherits FlexContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)

Name	Description
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>backgroundAlpha</i>	the alpha level of the color defined by the <i>backgroundColor</i> property, of the image or SWF file defined by the <i>backgroundImage</i> style. Valid values range from 0.0 to 1.0. (Inherited from <i>FlexContainer</i>)
<i>backgroundColor</i>	the background color of a component. (Inherited from <i>FlexContainer</i>)
<i>backgroundDisabledColor</i>	the background color of the component when it is disabled. The global default value is undefined. (Inherited from <i>FlexContainer</i>)
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class. (Inherited from <i>FlexContainer</i>)
<i>backgroundSize</i>	the percentage to change the image size to for the specified <i>backgroundImage</i> . (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)

Name	Description
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedChild</i>	a reference to the currently visible child container.
<i>selectedIndex</i>	the zero-based index of the currently visible child container.

Name	Description
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified

Name	Description
	interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FlexWindow Class

Description

The Window is a top-level container for additional windows in an AIR desktop application.

Inheritance Hierarchy

- [FlexContainer](#)
 - FlexWindow

Syntax

```
'Declaration
Public Class FlexWindow _
Inherits FlexContainer _
Implements IMoveable
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
alwaysInFront	a value whether the underlying NativeWindow is always in front of other windows.
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
backgroundAlpha	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from FlexContainer)
backgroundColor	the background color of a component. (Inherited from FlexContainer)
backgroundDisabledColor	the background color of the component when it is disabled. The global default value is undefined. (Inherited from FlexContainer)
backgroundImage	the background image of a component. This can be an absolute or relative URL or class. (Inherited from FlexContainer)

Name	Description
<i>backgroundSize</i>	the percentage to change the image size to for the specified backgroundImage. (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>closed</i>	a value whether the underlying window has been closed.
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>isAIRWindow</i>	a value whether this window is an AIR window.
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)

Name	Description
<i>maxHeight</i>	The maximum recommended height of the component to be considered by the parent during layout.
<i>maximizable</i>	a value whether the window can be maximized.
<i>maxWidth</i>	The maximum recommended width of the component to be considered by the parent during layout.
<i>minHeight</i>	The minimum recommended height of the component to be considered by the parent during layout.
<i>minimizable</i>	a value whether the window can be minimized.
<i>minWidth</i>	The minimum recommended width of the component to be considered by the parent during layout.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizable</i>	a value whether the window can be resized.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>showStatusBar</i>	a value whether the status bar is visible.
<i>status</i>	The string that appears in the status bar, if it is visible.
<i>systemChrome</i>	the type of system chrome (if any) the window has.

Name	Description
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>title</i>	The title that appears in the window title bar and the taskbar.
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>transparent</i>	a value whether the window is transparent.
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>WindowState</i>	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from <i>IMoveable</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)

Name	Description
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from IMoveable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StateChange	Dispatched when the displaystatus of the window is changed.
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

FlexWindowedApplication Class

Description

The `WindowedApplication` defines the application container that you use to create Flex applications for AIR applications.

Inheritance Hierarchy

- [FlexApplication](#)
 - FlexWindowedApplication

Syntax

```
'Declaration
Public Class FlexWindowedApplication _
Inherits FlexApplication _
Implements IMoveable
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>alwaysInFront</i>	a value whether the underlying NativeWindow is always in front of other windows.
<i>applicationID</i>	The identifier that AIR uses to identify the application.
<i>autoExit</i>	a value whether the AIR application will quit when the last window closes or will continue running in the background.
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>backgroundAlpha</i>	the alpha level of the color defined by the backgroundColor property, of the image or SWF file defined by the backgroundImage style. Valid values range from 0.0 to 1.0. (Inherited from <i>FlexContainer</i>)
<i>backgroundColor</i>	the background color of a component. (Inherited from <i>FlexContainer</i>)
<i>backgroundDisabledColor</i>	the background color of the component when it is disabled. The global default value is undefined. (Inherited from <i>FlexContainer</i>)
<i>backgroundImage</i>	the background image of a component. This can be an absolute or relative URL or class. (Inherited from <i>FlexContainer</i>)
<i>backgroundSize</i>	the percentage to change the image size to for the specified backgroundImage. (Inherited from <i>FlexContainer</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>closed</i>	a value whether the underlying window has been closed.
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>FlexContainer</i>)
<i>creationIndex</i>	the order to instantiate and draw the children of the container. (Inherited from <i>FlexContainer</i>)
<i>creationPolicy</i>	the child creation policy for this Container. (Inherited from <i>FlexContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)

Name	Description
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillAlphas</i>	the alpha transparency values used for the background fill of components. (Inherited from <i>FlexContainer</i>)
<i>fillColors</i>	the colors used to tint the background fill of the component. (Inherited from <i>FlexContainer</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>FlexContainer</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>fontFamily</i>	the name of the font to use. (Inherited from <i>FlexContainer</i>)
<i>fontSize</i>	the height of the text, in pixels. (Inherited from <i>FlexContainer</i>)
<i>fontStyle</i>	whether the text is italic font. (Inherited from <i>FlexContainer</i>)
<i>fontWeight</i>	whether the text is boldface. (Inherited from <i>FlexContainer</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text displayed by some navigator containers to represent this Container. (Inherited from <i>FlexContainer</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>FlexContainer</i>)
<i>maxHeight</i>	The maximum recommended height of the component to be considered by the parent during layout.
<i>maximizable</i>	a value whether the window can be maximized.
<i>maxWidth</i>	The maximum recommended width of the component to be considered by the parent during layout.
<i>minHeight</i>	The minimum recommended height of the component to be considered by the parent during layout.
<i>minimizable</i>	a value whether the window can be minimized.
<i>minWidth</i>	The minimum recommended width of the component to be considered by the parent during layout.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numChildren</i>	the number of child components in this container. (Inherited from <i>FlexContainer</i>)
<i>numColumns</i>	the total number of columns of data available. (Inherited from <i>FlexContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>FlexContainer</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizable</i>	a value whether the window can be resized.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>showStatusBar</i>	a value whether the status bar is visible.
<i>status</i>	The string that appears in the status bar, if it is visible.
<i>systemChrome</i>	the type of system chrome (if any) the window has.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TextObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>title</i>	The title that appears in the window title bar and the taskbar.
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>transparent</i>	a value whether the window is transparent.
<i>url</i>	the URL from which this Application's SWF file was loaded. (Inherited from <i>FlexApplication</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Defines the value of the type property of the event object for a dragComplete event. (Inherited from FlexContainer)
DragDrop	Defines the value of the type property of the event object for a dragDrop event. (Inherited from FlexContainer)
DragStart	Defines the value of the type property of the event object for a dragStart event. (Inherited from FlexContainer)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. The return value is an array of rows, each of which is an array of items. (Inherited from FlexContainer)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from FlexContainer)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from IMoveable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
Scroll	Dispatched when the user manually scrolls the container. (Inherited from FlexContainer)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkAirHTML Class

Description

The HTML control lets you display HTML content in your application.

Inheritance Hierarchy

- [FlexScrollBase](#)
 - SparkAirHTML

Syntax

```
'Declaration
Public Class SparkAirHTML _
Inherits FlexScrollBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)

Name	Description
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)

Name	Description
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from <i>FlexDisplayObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MouseScroll</i>	Dispatched when a mouse wheel is spun over an InteractiveObject instance. (Inherited from <i>FlexScrollBase</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the scroll events occur. This event is dispatched if the scrolling is done programmatically or by the user.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change

Name	Description
	the timeout by setting the value of the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkApplication Class

Description

Flex defines a default, or Application, container that lets you start adding content to your application without explicitly defining another container.

Inheritance Hierarchy

- [SparkSkinnableContainer](#)
 - SparkApplication
 - [SparkWindowedApplication](#)

Syntax

```
'Declaration
Public Class SparkApplication _
Inherits SparkSkinnableContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
colorCorrection	The value of the stage's colorCorrection property.
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableContainerBase)
controlBarVisible	whether the control bar is visible.
creationPolicy	the content creation policy for this component. (Inherited from SparkSkinnableContainer)
currentState	the current view state of the component. (Inherited from FlexObject)
deferredContentCreated	whether deferred content has been created. (Inherited from SparkSkinnableContainer)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from SparkSkinnableContainerBase)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
lastVisibleRow	the index of the last visible child. (Inherited from SparkSkinnableContainerBase)

Name	Description
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkSkinnableContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>url</i>	The URL from which this Application's SWF file was loaded.
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
viewSourceURL	the URL where the application's source can be viewed.
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkBorderContainer Class

Description

The `BorderContainer` class defines a set of CSS styles that control the appearance of the border and background fill of the container.

Inheritance Hierarchy

- [SparkSkinnableContainer](#)
 - `SparkBorderContainer`

Syntax

```
'Declaration  
Public Class SparkBorderContainer _  
Inherits SparkSkinnableContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableContainerBase)
creationPolicy	the content creation policy for this component. (Inherited from SparkSkinnableContainer)
currentState	the current view state of the component. (Inherited from FlexObject)
deferredContentCreated	whether deferred content has been created. (Inherited from SparkSkinnableContainer)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from SparkSkinnableContainerBase)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
lastVisibleRow	the index of the last visible child. (Inherited from SparkSkinnableContainerBase)

Name	Description
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkSkinnableContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)

Name	Description
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)

Name	Description
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkButton Class

Description

The `SparkButton` component is a commonly used rectangular button. The `SparkButton` component looks like it can be pressed.

Inheritance Hierarchy

- [SparkButtonBase](#)
 - `SparkButton`
 - [SparkMuteButton](#)

Syntax

```
'Declaration
Public Class SparkButton _
Inherits SparkButtonBase
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>autoRepeat</i>	whether to dispatch repeated buttonDown events if the user holds down the mouse button. (Inherited from <i>SparkButtonBase</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>emphasized</i>	whether the default button as requested by the focus manager.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text to appear on the ButtonBase control. (Inherited from <i>SparkButtonBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)

Name	Description
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>stickyHighlighting</i>	a value whether the button displays its down skin when the user presses it but changes to its over skin when the user drags the mouse off of it. (Inherited from <i>SparkButtonBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkButtonBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkButtonBar Class

Description

The ButtonBar control defines a horizontal group of logically related buttons with a common look and navigation.

Inheritance Hierarchy

- [SparkListBase](#)
 - SparkButtonBar

Syntax

```
'Declaration
Public Class SparkButtonBar _
Inherits SparkListBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	the measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableDataContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
caretIndex	The item that is currently in focus. (Inherited from SparkListBase)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableContainerBase)
currentState	the current view state of the component. (Inherited from FlexObject)

Name	Description
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>labelField</i>	The name of the field in the data provider items to display as the label. (Inherited from <i>SparkListBase</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>requireSelection</i>	a value if a data item must always be selected in the control (Inherited from <i>SparkListBase</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)

Name	Description
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	The 0-based index of the selected item, or -1 if no item is selected. (Inherited from <i>SparkListBase</i>)
<i>selectedItem</i>	The item that is currently selected. (Inherited from <i>SparkListBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>useVirtualLayout</i>	the value of the useVirtualLayout property of the layout associated with this control. (Inherited from <i>SparkListBase</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel

Name	Description
	must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemsCount	Gets the number of items in the ComboBox. (Inherited from SparkListBase)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Dispatched when an item in the list is selected. (Inherited from SparkListBase)
SelectIndex	Dispatched when the user clicks on an item in the list or navigates to the item using a keyboard.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 ToString 	Returns a string representation of the control. (Inherited from FlexDisplayObject)
 Type 	Dispatched when the user presses a key. (Inherited from SparkListBase)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by

Name	Description
WaitForPropertyNotEquals	the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkButtonBarButton Class

Description

The ButtonBarButton class defines the custom item renderer used by the ButtonBar control.

Inheritance Hierarchy

- [SparkToggleButton](#)
 - SparkButtonBarButton

Syntax

```
'Declaration
Public Class SparkButtonBarButton _
Inherits SparkToggleButton
```

Properties

Name	Description
allowDeselection	a value whether the user click on a currently selected button deselects it.
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)

Name	Description
<i>autoRepeat</i>	whether to dispatch repeated <code>buttonDown</code> events if the user holds down the mouse button. (Inherited from SparkButtonBase)
<i>className</i>	the name of this instance's class, such as <code>Button</code> . (Inherited from FlexObject)
<i>currentState</i>	the current view state of the component. (Inherited from FlexObject)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from FlexObject)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from FlexObject)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>label</i>	the text to appear on the <code>ButtonBase</code> control. (Inherited from SparkButtonBase)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is <code>true</code> , which means that by default any <code>InteractiveObject</code> instance that is on the display list receives mouse events. If <code>mouseEnabled</code> is set to <code>false</code> , the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the <code>mouseEnabled</code> behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is <code>NaN</code> . Setting the <code>width</code> or <code>explicitWidth</code> properties resets this property to <code>NaN</code> . (Inherited from FlexObject)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from FlexObject)

Name	Description
<i>selected</i>	whether the button is in the down state, and false if it is in the up state. (Inherited from <i>SparkToggleButtonBase</i>)
<i>showsCaret</i>	a value whether the item renderer can show itself as focused.
<i>stickyHighlighting</i>	a value whether the button displays its down skin when the user presses it but changes to its over skin when the user drags the mouse off of it. (Inherited from <i>SparkButtonBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	The event triggered when a drag operation is cancelled. (Inherited from SparkToggleButton)
DragDrop	The event triggered when the dragged item is dropped. (Inherited from SparkToggleButton)
DragStart	The event triggered when the drag/drop operation begins. (Inherited from SparkToggleButton)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)

Name	Description
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkButtonBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkButtonBase Class

Description

The SparkButtonBase class is the base class for the all Spark button components.

Inheritance Hierarchy

- [SparkObject](#)
 - SparkButtonBase
 - [SparkButton](#)
 - [SparkToggleButtonBase](#)

Syntax

```
'Declaration
Public Class SparkButtonBase _
Inherits SparkObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
autoRepeat	whether to dispatch repeated buttonDown events if the user holds down the mouse button.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)

Name	Description
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text to appear on the ButtonBase control.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>stickyHighlighting</i>	a value whether the button displays its down skin when the user presses it but changes to its over skin when the user drags the mouse off of it.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)

Name	Description
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change

Name	Description
	the timeout by setting the value of the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkCheckBox Class

Description

The `CheckBox` component consists of an optional label and a small box that can contain a check mark or not.

Inheritance Hierarchy

- [SparkToggleButtonBase](#)
 - `SparkCheckBox`

Syntax

```
'Declaration
Public Class SparkCheckBox _
Inherits SparkToggleButtonBase
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>autoRepeat</i>	whether to dispatch repeated buttonDown events if the user holds down the mouse button. (Inherited from <i>SparkButtonBase</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text to appear on the ButtonBase control. (Inherited from <i>SparkButtonBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)

Name	Description
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selected</i>	whether the button is in the down state, and false if it is in the up state. (Inherited from <i>SparkToggleButtonBase</i>)
<i>stickyHighlighting</i>	a value whether the button displays its down skin when the user presses it but changes to its over skin when the user drags the mouse off of it. (Inherited from <i>SparkButtonBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	The event triggered when a drag operation is cancelled.
DragDrop	The event triggered when the dragged item is dropped.
DragStart	The event triggered when the drag/drop operation begins.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkButtonBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkComboBox Class

Description

The ComboBox control is a child class of the DropDownListBase control. Like the DropDownListBase control, when the user selects an item from the drop-down list in the ComboBox control, the data item appears in the prompt area of the control.

Inheritance Hierarchy

- [SparkDropDownListBase](#)
 - SparkComboBox

Syntax

```
'Declaration
Public Class SparkComboBox _
Inherits SparkDropDownListBase
```

Properties

Name	Description
allowMultipleSelection	a value if multiple selection is enabled. (Inherited from SparkList)
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	the measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableDataContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)

Name	Description
<i>caretIndex</i>	The item that is currently in focus. (Inherited from <i>SparkListBase</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dragEnabled</i>	A flag that indicates whether you can drag items out of this control and drop them on other controls. (Inherited from <i>SparkList</i>)
<i>dragMoveEnabled</i>	A flag that indicates whether items can be moved instead of just copied from the control as part of a drag-and-drop operation. (Inherited from <i>SparkList</i>)
<i>dropEnabled</i>	A flag that indicates whether dragged items can be dropped onto the control. (Inherited from <i>SparkList</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>isDropDownOpen</i>	a value whether the drop down is open. (Inherited from <i>SparkDropDownListBase</i>)
<i>labelField</i>	The name of the field in the data provider items to display as the label. (Inherited from <i>SparkListBase</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>maxChars</i>	The maximum number of characters that the prompt area can contain, as entered by a user.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list

Name	Description
	are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available. (Inherited from SparkSkinnableContainerBase)
numRows	the total number of rows of data available. (Inherited from SparkSkinnableContainerBase)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
requireSelection	a value if a data item must always be selected in the control (Inherited from SparkListBase)
restrict	The set of characters that a user can enter into the prompt area.
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedIndex	The 0-based index of the selected item, or -1 if no item is selected. (Inherited from SparkListBase)
selectedIndices	a list representing the indices of the currently selected item or items. (Inherited from SparkList)
selectedItem	The item that is currently selected. (Inherited from SparkListBase)
selectedItems	a list of Objects representing the currently selected data items. (Inherited from SparkList)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)

Name	Description
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
useVirtualLayout	the value of the useVirtualLayout property of the layout associated with this control. (Inherited from SparkListBase)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemCount	Gets the number of items in the ComboBox. (Inherited from SparkListBase)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
<i>Input</i>	Dispatched when the user types, deletes, or pastes text into the control.
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>Open</i>	Dispatched when the user clicks the drop-down button to display the drop-down list. (Inherited from <i>SparkDropDownListBase</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Select</i>	Dispatched when an item in the list is selected. (Inherited from <i>SparkListBase</i>)
<i>SelectIndex</i>	Dispatched when the user clicks on an item in the list or navigates to the item using a keyboard. (Inherited from <i>SparkList</i>)
<i>SelectText</i>	Dispatched when text is selected.
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkListBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkComplexDisplay Class

Description

A complex rendered object.

Inheritance Hierarchy

- [SparkListLabel](#)
 - SparkComplexDisplay

Syntax

```
'Declaration
Public Class SparkComplexDisplay _
Inherits SparkListLabel
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	a value whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkGroupBase)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
automationValue	A value which can be used to uniquely identify this item. (Inherited from SparkListLabel)
blendMode	A value from the BlendMode class that specifies which blend mode to use. (Inherited from SparkGroup)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipAndEnableScrolling	a value whether to clip the children to the boundaries of the viewport. (Inherited from SparkGroupBase)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkGroupBase)

Name	Description
<i>contentHeight</i>	The height of the viewport's content. (Inherited from SparkGroupBase)
<i>contentWidth</i>	The width of the viewport's contents. (Inherited from SparkGroupBase)
<i>currentState</i>	the current view state of the component. (Inherited from FlexObject)
<i>dragging</i>	a value whether the SparkListLabel is in dragging mode. (Inherited from SparkListLabel)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from FlexObject)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from FlexObject)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from SparkGroupBase)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>horizontalScrollPosition</i>	The x coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component (Inherited from SparkGroupBase)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>label</i>	the label of the SparkListLabel. (Inherited from SparkListLabel)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from SparkGroupBase)
<i>luminosityClip</i>	a value whether the luminosity mask clips the masked content. (Inherited from SparkGroupBase)
<i>luminosityInvert</i>	a value that controls the calculation of the RGB color value of a graphic element being masked by a luminosity mask. (Inherited from SparkGroupBase)
<i>maskType</i>	The mask type. (Inherited from SparkGroupBase)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use

Name	Description
<i>mouseEnabledWhereTransparent</i>	flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>) a value whether the entire bounds of the Group respond to mouse events such as click and roll over. (Inherited from <i>SparkGroupBase</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkGroupBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkGroupBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkGroupBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizeMode</i>	The ResizeMode for this container. (Inherited from <i>SparkGroupBase</i>)
<i>scaleGridBottom</i>	the bottom coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridLeft</i>	the left coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridRight</i>	the right coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridTop</i>	the top coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selected</i>	a value whether the SparkListLabel is selected. (Inherited from <i>SparkListLabel</i>)
<i>showsCaret</i>	a value whether the caret is shown. (Inherited from <i>SparkListLabel</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalScrollPosition	The y coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component. (Inherited from SparkGroupBase)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkGroupBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkGroupBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkGroupBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkGroupBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when the mouse is used to move the scroll bars on this container. (Inherited from SparkGroupBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the container's scroll events occur. This event is dispatched if the scrolling is done programmatically or by the user. (Inherited from SparkGroupBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkGroupBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkDataGrid Class

Description

Identifies a Spark data grid control.

Inheritance Hierarchy

- [SparkSkinnableContainerBase](#)
 - SparkDataGrid

Syntax

```
'Declaration
Public Class SparkDataGrid _
Inherits SparkSkinnableContainerBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableContainerBase)
columnsLength	the count of the columns.
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)

Name	Description
firstVisibleRow	the index of the first visible child. (Inherited from SparkSkinnableContainerBase)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
lastVisibleRow	the index of the last visible child. (Inherited from SparkSkinnableContainerBase)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available. (Inherited from SparkSkinnableContainerBase)
numRows	the total number of rows of data available. (Inherited from SparkSkinnableContainerBase)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
rowHeight	the height of the row.
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedCell	the selected cell.
selectedCells	the selected cells.
selectedIndex	the index of the selected row.
selectedIndices	the indices of the selected rows.
selectedItem	the item of the selected row.
selectedItems	the items of the selected rows.

Name	Description
selectionLength	the number of selected rows.
selectionMode	the selection mode (singlerow, multiplerows, singlecells or multiplecells).
showDataTips	whether data tips should be shown.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
ColumnStretch	Stretches the given column of the data grid.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
EditNext	Sets the following data grid cell in the edit mode.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)

Name	Description
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HeaderClick	Clicks on the specified column header.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects the specified data grid cell.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter,

Name	Description
WaitForScreenshotStable	<p>the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

SparkDataGridLabel Class

Description

Identifies a spark data grid label control.

Inheritance Hierarchy

- [FlexObject](#)
 - SparkDataGridLabel

Syntax

```
'Declaration
Public Class SparkDataGridLabel _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
automationValue	A value which can be used to uniquely identify this item.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)

Name	Description
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)

Name	Description
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)

Name	Description
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By

Name	Description
	default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkDataGroup Class

Description

The DataGroup class is the base container class for data items.

Inheritance Hierarchy

- [SparkGroupBase](#)
 - SparkDataGroup

Syntax

```
'Declaration
Public Class SparkDataGroup _
Inherits SparkGroupBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	a value whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkGroupBase)

Name	Description
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>clipAndEnableScrolling</i>	a value whether to clip the children to the boundaries of the viewport. (Inherited from <i>SparkGroupBase</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkGroupBase</i>)
<i>contentHeight</i>	The height of the viewport's content. (Inherited from <i>SparkGroupBase</i>)
<i>contentWidth</i>	The width of the viewport's contents. (Inherited from <i>SparkGroupBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkGroupBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalScrollPosition</i>	The x coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component (Inherited from <i>SparkGroupBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkGroupBase</i>)
<i>luminosityClip</i>	a value whether the luminosity mask clips the masked content. (Inherited from <i>SparkGroupBase</i>)

Name	Description
<i>luminosityInvert</i>	a value that controls the calculation of the RGB color value of a graphic element being masked by a luminosity mask. (Inherited from <i>SparkGroupBase</i>)
<i>maskType</i>	The mask type. (Inherited from <i>SparkGroupBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>mouseEnabledWhereTransparent</i>	a value whether the entire bounds of the Group respond to mouse events such as click and roll over. (Inherited from <i>SparkGroupBase</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkGroupBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkGroupBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkGroupBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizeMode</i>	The ResizeMode for this container. (Inherited from <i>SparkGroupBase</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)

Name	Description
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalScrollPosition	The y coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component. (Inherited from SparkGroupBase)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkGroupBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkGroupBase)

Name	Description
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkGroupBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkGroupBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when the mouse is used to move the scroll bars on this container. (Inherited from SparkGroupBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the container's scroll events occur. This event is dispatched if the scrolling is done programmatically or by the user. (Inherited from SparkGroupBase)
Select	Dispatched when one or more items in the list are deselected.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkGroupBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkDataRenderer Class

Description

The SparkDataRenderer class is the base class for data components in spark.

Inheritance Hierarchy

- [SparkGroup](#)
 - SparkDataRenderer
 - [SparkListLabel](#)

Syntax

```
'Declaration
Public Class SparkDataRenderer _
Inherits SparkGroup
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	a value whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkGroupBase)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
blendMode	A value from the BlendMode class that specifies which blend mode to use. (Inherited from SparkGroup)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipAndEnableScrolling	a value whether to clip the children to the boundaries of the viewport. (Inherited from SparkGroupBase)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkGroupBase)
contentHeight	The height of the viewport's content. (Inherited from SparkGroupBase)

Name	Description
<i>contentWidth</i>	The width of the viewport's contents. (Inherited from <i>SparkGroupBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkGroupBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalScrollPosition</i>	The x coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component (Inherited from <i>SparkGroupBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkGroupBase</i>)
<i>luminosityClip</i>	a value whether the luminosity mask clips the masked content. (Inherited from <i>SparkGroupBase</i>)
<i>luminosityInvert</i>	a value that controls the calculation of the RGB color value of a graphic element being masked by a luminosity mask. (Inherited from <i>SparkGroupBase</i>)
<i>maskType</i>	The mask type. (Inherited from <i>SparkGroupBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>mouseEnabledWhereTransparent</i>	a value whether the entire bounds of the Group respond to mouse events such as click and roll over. (Inherited from <i>SparkGroupBase</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)

Name	Description
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkGroupBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkGroupBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkGroupBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizeMode</i>	The ResizeMode for this container. (Inherited from <i>SparkGroupBase</i>)
<i>scaleGridBottom</i>	the bottom coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridLeft</i>	the left coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridRight</i>	the right coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridTop</i>	the top coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalScrollPosition	The y coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component. (Inherited from SparkGroupBase)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkGroupBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkGroupBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkGroupBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from <i>FlexDisplayObject</i>)
<i>GetValues</i>	Returns a matrix containing the automation values of all parts of the components. (Inherited from <i>SparkGroupBase</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when the mouse is used to move the scroll bars on this container. (Inherited from SparkGroupBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the container's scroll events occur. This event is dispatched if the scrolling is done programmatically or by the user. (Inherited from SparkGroupBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkGroupBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkDropDownList Class

Description

The SparkDropDownList control contains a drop-down list from which the user can select a single value.

Inheritance Hierarchy

- [SparkDropDownListBase](#)
 - SparkDropDownList

Syntax

```
'Declaration  
Public Class SparkDropDownList _  
Inherits SparkDropDownListBase
```

Properties

Name	Description
allowMultipleSelection	a value if multiple selection is enabled. (Inherited from SparkList)
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	the measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableDataContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
caretIndex	The item that is currently in focus. (Inherited from SparkListBase)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableContainerBase)
currentState	the current view state of the component. (Inherited from FlexObject)
dragEnabled	A flag that indicates whether you can drag items out of this control and drop them on other controls. (Inherited from SparkList)
dragMoveEnabled	A flag that indicates whether items can be moved instead of just copied from the control as part of a drag-and-drop operation. (Inherited from SparkList)
dropEnabled	A flag that indicates whether dragged items can be dropped onto the control. (Inherited from SparkList)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)

Name	Description
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from SparkSkinnableContainerBase)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>isDropDownOpen</i>	a value whether the drop down is open. (Inherited from SparkDropDownListBase)
<i>labelField</i>	The name of the field in the data provider items to display as the label. (Inherited from SparkListBase)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from SparkSkinnableContainerBase)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from SparkSkinnableContainerBase)
<i>numRows</i>	the total number of rows of data available. (Inherited from SparkSkinnableContainerBase)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
<i>prompt</i>	The prompt for the DropDownList control.
<i>requireSelection</i>	a value if a data item must always be selected in the control (Inherited from SparkListBase)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)

Name	Description
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	The 0-based index of the selected item, or -1 if no item is selected. (Inherited from <i>SparkListBase</i>)
<i>selectedIndices</i>	a list representing the indices of the currently selected item or items. (Inherited from <i>SparkList</i>)
<i>selectedItem</i>	The item that is currently selected. (Inherited from <i>SparkListBase</i>)
<i>selectedItems</i>	a list of Objects representing the currently selected data items. (Inherited from <i>SparkList</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TextObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>useVirtualLayout</i>	the value of the useVirtualLayout property of the layout associated with this control. (Inherited from <i>SparkListBase</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TextObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemsCount	Gets the number of items in the ComboBox. (Inherited from SparkListBase)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
Open	Dispatched when the user clicks the drop-down button to display the drop-down list. (Inherited from SparkDropDownListBase)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Dispatched when an item in the list is selected. (Inherited from SparkListBase)
SelectIndex	Dispatched when the user clicks on an item in the list or navigates to the item using a keyboard. (Inherited from SparkList)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkListBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkDropDownListBase Class

Description

The `SparkDropDownListBase` control contains a drop-down list from which the user can select a single value.

Inheritance Hierarchy

- [SparkList](#)
 - `SparkDropDownListBase`
 - [SparkComboBox](#)
 - [SparkDropDownList](#)

Syntax

```
'Declaration
Public Class SparkDropDownListBase _
Inherits SparkList
```

Properties

Name	Description
<i>allowMultipleSelection</i>	a value if multiple selection is enabled. (Inherited from <i>SparkList</i>)
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>autoLayout</i>	the measurement and layout are done when the position or size of a child is changed. (Inherited from <i>SparkSkinnableDataContainer</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>caretIndex</i>	The item that is currently in focus. (Inherited from <i>SparkListBase</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dragEnabled</i>	A flag that indicates whether you can drag items out of this control and drop them on other controls. (Inherited from <i>SparkList</i>)
<i>dragMoveEnabled</i>	A flag that indicates whether items can be moved instead of just copied from the control as part of a drag-and-drop operation. (Inherited from <i>SparkList</i>)
<i>dropEnabled</i>	A flag that indicates whether dragged items can be dropped onto the control. (Inherited from <i>SparkList</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)

Name	Description
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>isDropDownOpen</i>	a value whether the drop down is open.
<i>labelField</i>	The name of the field in the data provider items to display as the label. (Inherited from SparkListBase)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from SparkSkinnableContainerBase)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from SparkSkinnableContainerBase)
<i>numRows</i>	the total number of rows of data available. (Inherited from SparkSkinnableContainerBase)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
<i>requireSelection</i>	a value if a data item must always be selected in the control (Inherited from SparkListBase)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
<i>selectedIndex</i>	The 0-based index of the selected item, or -1 if no item is selected. (Inherited from SparkListBase)
<i>selectedIndices</i>	a list representing the indices of the currently selected item or items. (Inherited from SparkList)
<i>selectedItem</i>	The item that is currently selected. (Inherited from SparkListBase)

Name	Description
selectedItems	a list of Objects representing the currently selected data items. (Inherited from SparkList)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
useVirtualLayout	the value of the useVirtualLayout property of the layout associated with this control. (Inherited from SparkListBase)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemsCount	Gets the number of items in the ComboBox. (Inherited from SparkListBase)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)

Name	Description
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
Open	Dispatched when the user clicks the drop-down button to display the drop-down list.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Dispatched when an item in the list is selected. (Inherited from SparkListBase)
SelectIndex	Dispatched when the user clicks on an item in the list or navigates to the item using a keyboard. (Inherited from SparkList)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 ToString 	Returns a string representation of the control. (Inherited from FlexDisplayObject)
 Type 	Dispatched when the user presses a key. (Inherited from SparkListBase)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by

Name	Description
WaitForPropertyNotEquals	the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkForm Class

Description

Identifies a Spark form control.

Inheritance Hierarchy

- [SparkSkinnableContainer](#)
 - SparkForm

Syntax

```
'Declaration
Public Class SparkForm _
Inherits SparkSkinnableContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)

Name	Description
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>creationPolicy</i>	the content creation policy for this component. (Inherited from <i>SparkSkinnableContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>deferredContentCreated</i>	whether deferred content has been created. (Inherited from <i>SparkSkinnableContainer</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkSkinnableContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)

Name	Description
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)</p>
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkFormItem Class

Description

Identifies a Spark form item control.

Inheritance Hierarchy

- [SparkSkinnableContainer](#)
 - SparkFormItem

Syntax

```
'Declaration
Public Class SparkFormItem _
Inherits SparkSkinnableContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableContainerBase)
creationPolicy	the content creation policy for this component. (Inherited from SparkSkinnableContainer)

Name	Description
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>deferredContentCreated</i>	whether deferred content has been created. (Inherited from <i>SparkSkinableContainer</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text of the form items label.
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinableContainerBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinableContainerBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkSkinableContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinableContainerBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth

Name	Description
	properties resets this property to NaN. (Inherited from FlexObject)
required	whether the form item has to have a value to submit the form.
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
sequenceLabel	the number of the form item in the form.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TextObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

SparkGroup Class

Description

Use this container when you want to manage visual children, both visual components and graphical components.

Inheritance Hierarchy

- [SparkGroupBase](#)
 - SparkGroup
 - [SparkDataRenderer](#)
 - [SparkTileGroup](#)

Syntax

```
'Declaration
Public Class SparkGroup _
Inherits SparkGroupBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	a value whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkGroupBase)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
blendMode	A value from the BlendMode class that specifies which blend mode to use.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)

Name	Description
<i>clipAndEnableScrolling</i>	a value whether to clip the children to the boundaries of the viewport. (Inherited from <i>SparkGroupBase</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkGroupBase</i>)
<i>contentHeight</i>	The height of the viewport's content. (Inherited from <i>SparkGroupBase</i>)
<i>contentWidth</i>	The width of the viewport's contents. (Inherited from <i>SparkGroupBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkGroupBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalScrollPosition</i>	The x coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component (Inherited from <i>SparkGroupBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkGroupBase</i>)
<i>luminosityClip</i>	a value whether the luminosity mask clips the masked content. (Inherited from <i>SparkGroupBase</i>)
<i>luminosityInvert</i>	a value that controls the calculation of the RGB color value of a graphic element being masked by a luminosity mask. (Inherited from <i>SparkGroupBase</i>)
<i>maskType</i>	The mask type. (Inherited from <i>SparkGroupBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use

Name	Description
	flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
mouseEnabledWhereTransparent	a value whether the entire bounds of the Group respond to mouse events such as click and roll over. (Inherited from SparkGroupBase)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available. (Inherited from SparkGroupBase)
numElements	The number of visual elements in this container. (Inherited from SparkGroupBase)
numRows	the total number of rows of data available. (Inherited from SparkGroupBase)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
resizeMode	The ResizeMode for this container. (Inherited from SparkGroupBase)
scaleGridBottom	the bottom coordinate of the scale grid.
scaleGridLeft	the left coordinate of the scale grid.
scaleGridRight	the right coordinate of the scale grid.
scaleGridTop	the top coordinate of the scale grid.
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)

Name	Description
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalScrollPosition	The y coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component. (Inherited from SparkGroupBase)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkGroupBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkGroupBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkGroupBase)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkGroupBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when the mouse is used to move the scroll bars on this container. (Inherited from SparkGroupBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Dispatched when the container's scroll events occur. This event is dispatched if the scrolling is done programmatically or by the user. (Inherited from SparkGroupBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkGroupBase)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkGroupBase Class

Description

The GroupBase class defines the base class for components that display visual elements.

Inheritance Hierarchy

- [FlexObject](#)
 - SparkGroupBase
 - [SparkDataGroup](#)
 - [SparkGroup](#)

Syntax

```
'Declaration  
Public Class SparkGroupBase _  
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	a value whether measurement and layout are done when the position or size of a child is changed.
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipAndEnableScrolling	a value whether to clip the children to the boundaries of the viewport.
columnNames	a list containing the names of all columns in the data.
contentHeight	The height of the viewport's content.
contentWidth	The width of the viewport's contents.
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)

Name	Description
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalScrollPosition</i>	The x coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child.
<i>luminosityClip</i>	a value whether the luminosity mask clips the masked content.
<i>luminosityInvert</i>	a value that controls the calculation of the RGB color value of a graphic element being masked by a luminosity mask.
<i>maskType</i>	The mask type.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>mouseEnabledWhereTransparent</i>	a value whether the entire bounds of the Group respond to mouse events such as click and roll over.
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available.
<i>numElements</i>	The number of visual elements in this container.
<i>numRows</i>	the total number of rows of data available.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizeMode</i>	The ResizeMode for this container.

Name	Description
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TextObject)
verticalScrollPosition	The y coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component.
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled.
DragDrop	Dispatched when the dragged item is dropped.
DragStart	Dispatched when the drag/drop operation begins.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components.
HighlightObject	Highlights this object. (Inherited from TestObject)

Name	Description
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>MouseScroll</i>	Dispatched when the mouse is used to move the scroll bars on this container.
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>Scroll</i>	Dispatched when the container's scroll events occur. This event is dispatched if the scrolling is done programmatically or by the user.
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from TestObject)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from TestObject)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from TestObject)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkImage Class

Description

Identifies a Spark image control.

Inheritance Hierarchy

- [SparkObject](#)
 - SparkImage

Syntax

```
'Declaration
Public Class SparkImage _
Inherits SparkObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
bytesLoaded	the number of bytes of the image already loaded.
bytesTotal	the total image data in bytes loaded or pending load.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clearOnLoad	whether or not to clear previous image content prior to loading new content.
contentLoaderGrouping	the optional content grouping identifier to pass to the an associated <code>IContentLoader</code> instance's <code>load()</code> method.
currentState	the current view state of the component. (Inherited from FlexObject)

Name	Description
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>fillMode</i>	a value how the bitmap fills in the dimensions.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalAlign</i>	the horizontal alignment of the content when it does not have a one-to-one aspect ratio and <i>scaleMode</i> is set to <i>BitmapScaleMode.LETTERBOX</i> .
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any <i>InteractiveObject</i> instance that is on the display list receives mouse events. If <i>mouseEnabled</i> is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the <i>mouseEnabled</i> behavior for all children of an object on the display list, use <i>flash.display.DisplayObjectContainer.mouseChildren</i> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or <i>explicitWidth</i> properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>preliminaryHeight</i>	an estimate to use for height when the "measured" bounds of the image is requested by layout, but the image data has yet to complete loading.
<i>preliminaryWidth</i>	an estimate to use for width when the "measured" bounds of the image is requested by layout, but the image data has yet to complete loading.
<i>scaleMode</i>	whether the image is scaled when <i>fillMode</i> is set to <i>BitmapFillMode.SCALE</i> .
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)

Name	Description
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>smooth</i>	whether to apply a smoothing algorithm to the bitmap image.
<i>source</i>	the content location.
<i>sourceHeight</i>	the unscaled height of the original image data.
<i>sourceWidth</i>	the unscaled width of the original image data.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>trustedSource</i>	a flag denoting whether the currently loaded content is considered loaded from a source whose security policy allows for cross domain image access.
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>verticalAlign</i>	the vertical alignment of the content when it does not have a one-to-one aspect ratio and scaleMode is set to BitmapScaleMode.LETTERBOX.
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkLabel Class

Description

Label is a low-level UIComponent that can render one or more lines of uniformly-formatted text.

Inheritance Hierarchy

- [SparkTextBase](#)
 - SparkLabel

Syntax

```
'Declaration
Public Class SparkLabel _
Inherits SparkTextBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)

Name	Description
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>isTruncated</i>	whether the text has been truncated. (Inherited from SparkTextBase)
<i>maxDisplayedLines</i>	An integer which determines whether, and where, the text gets truncated. (Inherited from SparkTextBase)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from FlexDisplayObject)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
<i>Text</i>	The text of the control. (Inherited from TextObject)
<i>text</i>	The text displayed by this text component. (Inherited from SparkTextBase)
<i>themeColor</i>	the theme color of a component. (Inherited from FlexObject)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is

Name	Description
	false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	The event triggered when a drag operation is cancelled. (Inherited from SparkTextBase)
DragDrop	The event triggered when the dragged item is dropped. (Inherited from SparkTextBase)
DragStart	The event triggered when the drag/drop operation begins. (Inherited from SparkTextBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkTextBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkList Class

Description

The List control displays a vertical list of items.

Inheritance Hierarchy

- [SparkListBase](#)
 - SparkList
 - [SparkDropDownListBase](#)

Syntax

```
'Declaration
Public Class SparkList _
Inherits SparkListBase
```


Properties

Name	Description
<i>allowMultipleSelection</i>	a value if multiple selection is enabled.
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>autoLayout</i>	the measurement and layout are done when the position or size of a child is changed. (Inherited from <i>SparkSkinnableDataContainer</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>caretIndex</i>	The item that is currently in focus. (Inherited from <i>SparkListBase</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dragEnabled</i>	A flag that indicates whether you can drag items out of this control and drop them on other controls.
<i>dragMoveEnabled</i>	A flag that indicates whether items can be moved instead of just copied from the control as part of a drag-and-drop operation.
<i>dropEnabled</i>	A flag that indicates whether dragged items can be dropped onto the control.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)

Name	Description
<i>labelField</i>	The name of the field in the data provider items to display as the label. (Inherited from SparkListBase)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from SparkSkinnableContainerBase)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from SparkSkinnableContainerBase)
<i>numRows</i>	the total number of rows of data available. (Inherited from SparkSkinnableContainerBase)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
<i>requireSelection</i>	a value if a data item must always be selected in the control (Inherited from SparkListBase)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
<i>selectedIndex</i>	The 0-based index of the selected item, or -1 if no item is selected. (Inherited from SparkListBase)
<i>selectedIndices</i>	a list representing the indices of the currently selected item or items.
<i>selectedItem</i>	The item that is currently selected. (Inherited from SparkListBase)
<i>selectedItems</i>	a list of Objects representing the currently selected data items.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from FlexDisplayObject)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
useVirtualLayout	the value of the useVirtualLayout property of the layout associated with this control. (Inherited from SparkListBase)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinableContainerBase)

Name	Description
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemsCount	Gets the number of items in the ComboBox. (Inherited from SparkListBase)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Dispatched when an item in the list is selected. (Inherited from SparkListBase)
SelectIndex	Dispatched when the user clicks on an item in the list or navigates to the item using a keyboard.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkListBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkListBase Class

Description

The ListBase class is the base class for all components that support selection.

Inheritance Hierarchy

- [SparkSkinnableDataContainer](#)
 - SparkListBase
 - [SparkButtonBar](#)
 - [SparkList](#)
 - [SparkTabBar](#)

Syntax

```
'Declaration
Public Class SparkListBase _
Inherits SparkSkinnableDataContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	the measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableDataContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
caretIndex	The item that is currently in focus.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableContainerBase)

Name	Description
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>labelField</i>	The name of the field in the data provider items to display as the label.
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>requireSelection</i>	a value if a data item must always be selected in the control

Name	Description
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectedIndex	The 0-based index of the selected item, or -1 if no item is selected.
selectedItem	The item that is currently selected.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
useVirtualLayout	the value of the useVirtualLayout property of the layout associated with this control.
Value	The value of the control, e.g.: text in a text control. (Inherited from TextObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The

Name	Description
	<p>captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemsCount	Gets the number of items in the ComboBox.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Dispatched when an item in the list is selected.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)

Name	Description
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkListLabel Class

Description

A simple rendered object.

Inheritance Hierarchy

- [SparkDataRenderer](#)
 - SparkListLabel
 - [SparkComplexDisplay](#)

Syntax

```
'Declaration
Public Class SparkListLabel _
Inherits SparkDataRenderer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	a value whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkGroupBase)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
automationValue	A value which can be used to uniquely identify this item.
blendMode	A value from the BlendMode class that specifies which blend mode to use. (Inherited from SparkGroup)

Name	Description
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>clipAndEnableScrolling</i>	a value whether to clip the children to the boundaries of the viewport. (Inherited from <i>SparkGroupBase</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkGroupBase</i>)
<i>contentHeight</i>	The height of the viewport's content. (Inherited from <i>SparkGroupBase</i>)
<i>contentWidth</i>	The width of the viewport's contents. (Inherited from <i>SparkGroupBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>dragging</i>	a value whether the SparkListLabel is in dragging mode.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkGroupBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>horizontalScrollPosition</i>	The x coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component (Inherited from <i>SparkGroupBase</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the label of the SparkListLabel.
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkGroupBase</i>)
<i>luminosityClip</i>	a value whether the luminosity mask clips the masked content. (Inherited from <i>SparkGroupBase</i>)
<i>luminosityInvert</i>	a value that controls the calculation of the RGB color value of a graphic element being masked by a luminosity mask. (Inherited from <i>SparkGroupBase</i>)
<i>maskType</i>	The mask type. (Inherited from <i>SparkGroupBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject

Name	Description
	instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>mouseEnabledWhereTransparent</i>	a value whether the entire bounds of the Group respond to mouse events such as click and roll over. (Inherited from <i>SparkGroupBase</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkGroupBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkGroupBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkGroupBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>resizeMode</i>	The ResizeMode for this container. (Inherited from <i>SparkGroupBase</i>)
<i>scaleGridBottom</i>	the bottom coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridLeft</i>	the left coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridRight</i>	the right coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridTop</i>	the top coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selected</i>	a value whether the SparkListLabel is selected.
<i>showsCaret</i>	a value whether the caret is shown.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
verticalScrollPosition	The y coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component. (Inherited from SparkGroupBase)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkGroupBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkGroupBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkGroupBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkGroupBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MouseScroll</i>	Dispatched when the mouse is used to move the scroll bars on this container. (Inherited from <i>SparkGroupBase</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Scroll</i>	Dispatched when the container's scroll events occur. This event is dispatched if the scrolling is done programmatically or by the user. (Inherited from <i>SparkGroupBase</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkGroupBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkMuteButton Class

Description

The MuteButton class defines the mute button used by the VideoPlayer control.

Inheritance Hierarchy

- [SparkButton](#)
 - SparkMuteButton

Syntax

'Declaration

Public Class SparkMuteButton _

Inherits SparkButton

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
autoRepeat	whether to dispatch repeated buttonDown events if the user holds down the mouse button. (Inherited from SparkButtonBase)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
emphasized	whether the default button as requested by the focus manager. (Inherited from SparkButton)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)

Name	Description
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text to appear on the ButtonBase control. (Inherited from <i>SparkButtonBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>muted</i>	a value whether the control is muted.
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>stickyHighlighting</i>	a value whether the button displays its down skin when the user presses it but changes to its over skin when the user drags the mouse off of it. (Inherited from <i>SparkButtonBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>volume</i>	the volume of the control.
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from <i>FlexDisplayObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
MuteChange	Toggles the muting of the control.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkButtonBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
WaitForObject	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

SparkNavigatorContent Class

Description

The `NavigatorContent` class defines a Spark container that can be used in an MX navigator container, such as the `ViewStack`, `TabNavigator` and `Accordion` containers.

Inheritance Hierarchy

- [SparkSkinnableContainer](#)
 - `SparkNavigatorContent`

Syntax

```
'Declaration
Public Class SparkNavigatorContent _
Inherits SparkSkinnableContainer
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>autoLayout</i>	whether measurement and layout are done when the position or size of a child is changed. (Inherited from <i>SparkSkinnableContainer</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>creationPolicy</i>	the content creation policy for this component. (Inherited from <i>SparkSkinnableContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>deferredContentCreated</i>	whether deferred content has been created. (Inherited from <i>SparkSkinnableContainer</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any <i>InteractiveObject</i> instance that is on the display list receives mouse events. If <i>mouseEnabled</i> is set to false, the instance does not receive any

Name	Description
	mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available. (Inherited from SparkSkinnableContainerBase)
numElements	The number of visual elements in this container. (Inherited from SparkSkinnableContainer)
numRows	the total number of rows of data available. (Inherited from SparkSkinnableContainerBase)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)

Name	Description
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkNumericStepper Class

Description

The `NumericStepper` control lets you select a number from an ordered set. The `NumericStepper` provides the same functionality as the `Spinner` component, but adds a `TextInput` control so that you can directly edit the value of the component, rather than modifying it by using the control's arrow buttons.

Inheritance Hierarchy

- [SparkSpinner](#)
 - `SparkNumericStepper`

Syntax

```
'Declaration
Public Class SparkNumericStepper _
Inherits SparkSpinner
```

Properties

Name	Description
<i>allowValueWrap</i>	the behavior of the control for a step when the current value is either the maximum or minimum value. (Inherited from <i>SparkSpinner</i>)
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>enableIME</i>	a value whether the IME should be enabled when the component receives focus.
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>imeMode</i>	the IME (Input Method Editor) mode.
<i>maxChars</i>	The maximum number of characters that can be entered in the field.
<i>maximum</i>	The maximum valid value. (Inherited from <i>SparkRange</i>)
<i>minimum</i>	The minimum valid value. (Inherited from <i>SparkRange</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use

Name	Description
	flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
snapInterval	the snapInterval. (Inherited from SparkRange)
stepSize	The amount that the value property changes when the changeValueByStep() method is called. (Inherited from SparkRange)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
value	The current value for this range. (Inherited from SparkRange)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)

Name	Description
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when the value of the NumericStepper changes as a result of user interaction. (Inherited from SparkSpinner)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Input	Dispatched when the user types, deletes, or pastes text into the control.
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
SelectText	Dispatched when text is selected.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkSpinner)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkObject Class

Description

The SparkObject class defines the base class for all spark components.

Inheritance Hierarchy

- [FlexObject](#)
 - SparkObject
 - [SparkButtonBase](#)
 - [SparkImage](#)
 - [SparkRange](#)
 - [SparkRichEditableText](#)
 - [SparkSkinnableContainerBase](#)
 - [SparkSkinnableTextBase](#)
 - [SparkVideoPlayer](#)

Syntax

```
'Declaration
Public Class SparkObject _
Inherits FlexObject
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)

Name	Description
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>ToString</i>	Returns a string representation of the control. (Inherited from <i>FlexDisplayObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkPanel Class

Description

The Panel class defines a container that includes a title bar, a caption, a border, and a content area for its children.

Inheritance Hierarchy

- [SparkSkinnableContainer](#)
 - SparkPanel
 - [SparkTitleWindow](#)

Syntax

```
'Declaration  
Public Class SparkPanel _  
Inherits SparkSkinnableContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableContainerBase)
controlBarVisible	a value whether the control bar is visible.
creationPolicy	the content creation policy for this component. (Inherited from SparkSkinnableContainer)
currentState	the current view state of the component. (Inherited from FlexObject)
deferredContentCreated	whether deferred content has been created. (Inherited from SparkSkinnableContainer)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from SparkSkinnableContainerBase)

Name	Description
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkSkinnableContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)

Name	Description
<i>title</i>	the title or caption displayed in the title bar.
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>DragCancel</i>	Dispatched when a drag operation is cancelled. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>DragDrop</i>	Dispatched when the dragged item is dropped. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>DragStart</i>	Dispatched when the drag/drop operation begins. (Inherited from <i>SparkSkinnableContainerBase</i>)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from <i>FlexObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>ToString</i>	Returns a string representation of the control. (Inherited from <i>FlexDisplayObject</i>)
<i>Type</i>	Dispatched when the user presses a key.
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkPopUpAnchor Class

Description

Identifies a Spark PopUpAnchor control.

Inheritance Hierarchy

- [FlexObject](#)
 - SparkPopUpAnchor

Syntax

```
'Declaration  
Public Class SparkPopUpAnchor _  
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
displayPopUp	whether the popUp is visible.
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)

Name	Description
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>popUpHeightMatchesAnchorHeight</i>	whether the height of the popUp control is set to the value of the PopUpAnchor's height.
<i>popUpPosition</i>	the position of the popUp control when it is opened, relative to the PopUpAnchor component.
<i>popUpWidthMatchesAnchorWidth</i>	whether the width of the popUp control is set to the value of the PopUpAnchor's width.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkRadioButton Class

Description

The RadioButton component allows the user make a single choice within a set of mutually exclusive choices. A RadioButtonGroup is composed of two or more RadioButton components with the same groupName property.

Inheritance Hierarchy

- [SparkToggleButtonBase](#)
 - SparkRadioButton

Syntax

```
'Declaration
Public Class SparkRadioButton _
Inherits SparkToggleButtonBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
autoRepeat	whether to dispatch repeated buttonDown events if the user holds down the mouse button. (Inherited from SparkButtonBase)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)

Name	Description
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>groupName</i>	the name of the group to which this RadioButton component belongs, or specifies the value of the id property of a RadioButtonGroup component if this RadioButton is part of a group defined by a RadioButtonGroup component.
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text to appear on the ButtonBase control. (Inherited from <i>SparkButtonBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selected</i>	whether the button is in the down state, and false if it is in the up state. (Inherited from <i>SparkToggleButtonBase</i>)

Name	Description
stickyHighlighting	a value whether the button displays its down skin when the user presses it but changes to its over skin when the user drags the mouse off of it. (Inherited from SparkButtonBase)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	The event triggered when a drag operation is cancelled.
DragDrop	The event triggered when the dragged item is dropped.
DragStart	The event triggered when the drag/drop operation begins.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)

Name	Description
Type	Dispatched when the user presses a key. (Inherited from SparkButtonBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkRange Class

Description

The Range class holds a value and an allowed range for that value, defined by minimum and maximum properties.

Inheritance Hierarchy

- [SparkObject](#)
 - SparkRange
 - [SparkSpinner](#)
 - [SparkTrackBase](#)

Syntax

```
'Declaration  
Public Class SparkRange _  
Inherits SparkObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)

Name	Description
<i>maximum</i>	The maximum valid value.
<i>minimum</i>	The minimum valid value.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>snapInterval</i>	the snapInterval.
<i>stepSize</i>	The amount that the value property changes when the changeValueByStep() method is called.
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)

Name	Description
<i>value</i>	The current value for this range.
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from <i>FlexDisplayObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkRichEditableText Class

Description

RichEditableText is a low-level UIComponent for displaying, scrolling, selecting, and editing richly-formatted text.

Inheritance Hierarchy

- [SparkObject](#)
 - SparkRichEditableText

Syntax

```
'Declaration
Public Class SparkRichEditableText _
Inherits SparkObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)

Name	Description
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>clipAndEnableScrolling</i>	whether to clip the children to the boundaries of the viewport.
<i>columnNames</i>	a list containing the names of all columns in the data.
<i>contentHeight</i>	The height of the text.
<i>contentWidth</i>	The width of the text.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>displayAsPassword</i>	whether the text field is a password text field.
<i>editable</i>	A flag indicating whether the user is allowed to edit the text in this control.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>enableIME</i>	A flag that indicates whether the IME should be enabled when the component receives focus.
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>heightInLines</i>	The default height of the control, measured in lines.
<i>horizontalScrollPosition</i>	The number of pixels by which the text is scrolled horizontally.
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>imeMode</i>	the IME (input method editor) mode.
<i>lastVisibleRow</i>	the index of the last visible child.
<i>maxChars</i>	The maximum number of characters that the text field can contain, as entered by a user.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use

Name	Description
	flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
multiline	whether the user can enter multiline text.
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available.
numRows	the total number of rows of data available.
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
restrict	the set of characters that a user can enter into the text field.
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectable	A flag indicating whether the content is selectable with the mouse, or with the keyboard when the control has the keyboard focus.
selectionActivePosition	A character position, relative to the beginning of the text String, specifying the end of the selection that moves when the selection is extended with the arrow keys.
selectionAnchorPosition	A character position, relative to the beginning of the text String, specifying the end of the selection that stays fixed when the selection is extended with the arrow keys.
selectionHighlighting	a value whether the text selection is highlighted.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
text	The text String displayed by this component.
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)

Name	Description
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If <i>useHandCursor</i> is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>verticalScrollPosition</i>	The number of pixels by which the text is scrolled vertically.
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>widthInChars</i>	The default width of the control, measured in em units.
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>DragCancel</i>	The event triggered when a drag operation is cancelled.
<i>DragDrop</i>	The event triggered when the dragged item is dropped.
<i>DragStart</i>	The event triggered when the drag/drop operation begins.
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from <i>FlexDisplayObject</i>)
<i>GetValues</i>	Returns a matrix containing the automation values of all parts of the components.
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Input</i>	Dispatched when the user types, deletes, or pastes text into the control.
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when the mouse is used to move the scroll bars on this container.
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SelectText	Dispatched when text is selected.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkRichText Class

Description

RichText is a low-level UIComponent that can display one or more lines of richly-formatted text and embedded images.

Inheritance Hierarchy

- [SparkTextBase](#)
 - SparkRichText

Syntax

```
'Declaration  
Public Class SparkRichText _  
Inherits SparkTextBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
isTruncated	whether the text has been truncated. (Inherited from SparkTextBase)
luminosityClip	whether the luminosity mask clips the masked content.
luminosityInvert	whether the calculation of the RGB color value of a graphic element is masked by a luminosity mask.
maskType	how the mask is applied to the component. The possible values are <code>MaskType.CLIP</code> , <code>MaskType.ALPHA</code> and <code>MaskType.LUMINOSITY</code> . Clip Masking When masking in clip mode, a clipping masks is reduced to 1-bit.

Name	Description
<i>maxDisplayedLines</i>	An integer which determines whether, and where, the text gets truncated. (Inherited from <i>SparkTextBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>text</i>	The text displayed by this text component. (Inherited from <i>SparkTextBase</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>DragCancel</i>	The event triggered when a drag operation is cancelled. (Inherited from <i>SparkTextBase</i>)
<i>DragDrop</i>	The event triggered when the dragged item is dropped. (Inherited from <i>SparkTextBase</i>)
<i>DragStart</i>	The event triggered when the drag/drop operation begins. (Inherited from <i>SparkTextBase</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkTextBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkScrollBar Class

Description

The `ScrollBarBase` class helps to position the portion of data that is displayed when there is too much data to fit in a display area.

Inheritance Hierarchy

- [SparkTrackBase](#)
 - SparkScrollBar

Syntax

```
'Declaration
Public Class SparkScrollBar _
Inherits SparkTrackBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)

Name	Description
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>maximum</i>	The maximum valid value. (Inherited from <i>SparkRange</i>)
<i>minimum</i>	The minimum valid value. (Inherited from <i>SparkRange</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>pageSize</i>	The change in the value of the value property when you call the <code>changeValueByPage()</code> method.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or <code>explicitWidth</code>

Name	Description
	properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
snapInterval	the snapInterval. (Inherited from SparkRange)
stepSize	The amount that the value property changes when the changeValueByStep() method is called. (Inherited from SparkRange)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TextObject)
value	The current value for this range. (Inherited from SparkRange)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when the value of the NumericStepper changes as a result of user interaction. (Inherited from SparkTrackBase)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkTrackBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkSkinnableContainer Class

Description

The SparkSkinnableContainer class is the base class for skinnable containers that have visual content.

Inheritance Hierarchy

- [SparkSkinnableContainerBase](#)
 - SparkSkinnableContainer
 - [SparkApplication](#)
 - [SparkBorderContainer](#)
 - [SparkForm](#)
 - [SparkFormItem](#)
 - [SparkNavigatorContent](#)
 - [SparkPanel](#)
 - [SparkSkinnablePopUpContainer](#)
 - [SparkWindow](#)

Syntax

```
'Declaration
Public Class SparkSkinnableContainer _
Inherits SparkSkinnableContainerBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	whether measurement and layout are done when the position or size of a child is changed.
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)

Name	Description
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>creationPolicy</i>	the content creation policy for this component.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>deferredContentCreated</i>	whether deferred content has been created.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numElements</i>	The number of visual elements in this container.
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)

Name	Description
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

SparkSkinnableContainerBase Class

Description

Base class for spark container components.

Inheritance Hierarchy

- [SparkObject](#)
 - SparkSkinnableContainerBase
 - [SparkDataGrid](#)
 - [SparkSkinnableContainer](#)
 - [SparkSkinnableDataContainer](#)

Syntax

'Declaration

```
Public Class SparkSkinnableContainerBase _
```

```
Inherits SparkObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data.
currentState	the current view state of the component. (Inherited from FlexObject)

Name	Description
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available.
<i>numRows</i>	the total number of rows of data available.
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled.
DragDrop	Dispatched when the dragged item is dropped.
DragStart	Dispatched when the drag/drop operation begins.

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForProperty	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkSkinnableDataContainer Class

Description

The SkinnableDataContainer class is the base container class for data items.

Inheritance Hierarchy

- [SparkSkinnableContainerBase](#)
 - SparkSkinnableDataContainer

- [SparkListBase](#)

Syntax

'Declaration

```
Public Class SparkSkinnableDataContainer _
```

```
Inherits SparkSkinnableContainerBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	the measurement and layout are done when the position or size of a child is changed.
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableContainerBase)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from SparkSkinnableContainerBase)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
lastVisibleRow	the index of the last visible child. (Inherited from SparkSkinnableContainerBase)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If

Name	Description
	mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available. (Inherited from SparkSkinnableContainerBase)
numRows	the total number of rows of data available. (Inherited from SparkSkinnableContainerBase)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)

Name	Description
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this <code>TestObject</code> . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this <code>TestObject</code> . (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By

Name	Description
	default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkSkinnablePopUpContainer Class

Description

Identifies a SparkSkinnablePopUpContainer control.

Inheritance Hierarchy

- [SparkSkinnableContainer](#)
 - SparkSkinnablePopUpContainer

Syntax

```
'Declaration
Public Class SparkSkinnablePopUpContainer _
Inherits SparkSkinnableContainer
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableContainer)

Name	Description
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>creationPolicy</i>	the content creation policy for this component. (Inherited from <i>SparkSkinnableContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>deferredContentCreated</i>	whether deferred content has been created. (Inherited from <i>SparkSkinnableContainer</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>isOpen</i>	whether the container is open.
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any <i>InteractiveObject</i> instance that is on the display list receives mouse events. If <i>mouseEnabled</i> is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the <i>mouseEnabled</i> behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)

Name	Description
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkSkinnableContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from <i>FlexDisplayObject</i>)
<i>GetValues</i>	Returns a matrix containing the automation values of all parts of the components. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from <i>FlexObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkSkinnableTextBase Class

Description

The base class for skinnable components, such as the Spark TextInput and TextArea, that include an instance of RichEditableText in their skin to provide rich text display, scrolling, selection, and editing.

Inheritance Hierarchy

- [SparkObject](#)
 - SparkSkinnableTextBase
 - [SparkTextArea](#)
 - [SparkTextInput](#)

Syntax

```
'Declaration
Public Class SparkSkinnableTextBase _
Inherits SparkObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)

Name	Description
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data.
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>displayAsPassword</i>	whether the text field is a password text field.
<i>editable</i>	A flag indicating whether the user is allowed to edit the text in this control.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>enableIME</i>	A flag that indicates whether the IME should be enabled when the component receives focus.
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child.
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>imeMode</i>	the IME (input method editor) mode.
<i>lastVisibleRow</i>	the index of the last visible child.
<i>maxChars</i>	The maximum number of characters that the text field can contain, as entered by a user.
<i>maxWidth</i>	The maximum recommended width of the component to be considered by the parent during layout.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use

Name	Description
	flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available.
numRows	the total number of rows of data available.
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
restrict	the set of characters that a user can enter into the text field.
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectable	A flag indicating whether the content is selectable with the mouse, or with the keyboard when the control has the keyboard focus.
selectionActivePosition	A character position, relative to the beginning of the text String, specifying the end of the selection that moves when the selection is extended with the arrow keys.
selectionAnchorPosition	A character position, relative to the beginning of the text String, specifying the end of the selection that stays fixed when the selection is extended with the arrow keys.
selectionHighlighting	a value whether the text selection is highlighted.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
text	The text String displayed by this component.
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is

Name	Description
	false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Input	Dispatched when the user types, deletes, or pastes text into the control.
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject . (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when the mouse is used to move the scroll bars on this container.

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SelectText	Dispatched when text is selected.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkSlider Class

Description

Lets users select a value by moving a slider thumb between the end points of the slider track.

Inheritance Hierarchy

- [SparkTrackBase](#)
 - SparkSlider
 - [SparkVolumeBar](#)

Syntax

```
'Declaration
Public Class SparkSlider _
Inherits SparkTrackBase
```


Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>maximum</i>	The maximum valid value. (Inherited from <i>SparkRange</i>)
<i>minimum</i>	The minimum valid value. (Inherited from <i>SparkRange</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth

Name	Description
	properties resets this property to NaN. (Inherited from FlexObject)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
snapInterval	the snapInterval. (Inherited from SparkRange)
stepSize	The amount that the value property changes when the changeValueByStep() method is called. (Inherited from SparkRange)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
value	The current value for this range. (Inherited from SparkRange)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when the value of the NumericStepper changes as a result of user interaction. (Inherited from SparkTrackBase)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkTrackBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkSpinner Class

Description

A Spinner component selects a value from an ordered set. It uses two buttons that increase or decrease the current value based on the current value of the stepSize property.

Inheritance Hierarchy

- [SparkRange](#)
 - SparkSpinner
 - [SparkNumericStepper](#)

Syntax

```
'Declaration
Public Class SparkSpinner _
Inherits SparkRange
```

Properties

Name	Description
allowValueWrap	the behavior of the control for a step when the current value is either the maximum or minimum value.
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)

Name	Description
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>maximum</i>	The maximum valid value. (Inherited from <i>SparkRange</i>)
<i>minimum</i>	The minimum valid value. (Inherited from <i>SparkRange</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>snapInterval</i>	the snapInterval. (Inherited from <i>SparkRange</i>)
<i>stepSize</i>	The amount that the value property changes when the changeValueByStep() method is called. (Inherited from <i>SparkRange</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
value	The current value for this range. (Inherited from SparkRange)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when the value of the NumericStepper changes as a result of user interaction.
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using

Name	Description
	the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkTabBar Class

Description

The TabBar class displays a set of identical tabs. One tab can be selected at a time, and the first tab is selected by default.

Inheritance Hierarchy

- [SparkListBase](#)

- SparkTabBar

Syntax

```
'Declaration
Public Class SparkTabBar _
Inherits SparkListBase
```

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>autoLayout</i>	the measurement and layout are done when the position or size of a child is changed. (Inherited from <i>SparkSkinnableDataContainer</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>caretIndex</i>	The item that is currently in focus. (Inherited from <i>SparkListBase</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>labelField</i>	The name of the field in the data provider items to display as the label. (Inherited from <i>SparkListBase</i>)

Name	Description
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>requireSelection</i>	a value if a data item must always be selected in the control (Inherited from <i>SparkListBase</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectedIndex</i>	The 0-based index of the selected item, or -1 if no item is selected. (Inherited from <i>SparkListBase</i>)
<i>selectedItem</i>	The item that is currently selected. (Inherited from <i>SparkListBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)

Name	Description
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
useVirtualLayout	the value of the useVirtualLayout property of the layout associated with this control. (Inherited from SparkListBase)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemsCount	Gets the number of items in the ComboBox. (Inherited from SparkListBase)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Dispatched when an item in the list is selected. (Inherited from SparkListBase)
SelectIndex	Dispatched when the user clicks on an item in the list or navigates to the item using a keyboard.
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkListBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkTextArea Class

Description

TextArea is a text-entry control that lets users enter and edit multiple lines of richly formatted text. It can display horizontal and vertical scrollbars for scrolling through the text and supports vertical scrolling with the mouse wheel.

Inheritance Hierarchy

- [SparkSkinnableTextBase](#)
 - SparkTextArea

Syntax

```
'Declaration
Public Class SparkTextArea _
Inherits SparkSkinnableTextBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableTextBase)
currentState	the current view state of the component. (Inherited from FlexObject)
displayAsPassword	whether the text field is a password text field. (Inherited from SparkSkinnableTextBase)
editable	A flag indicating whether the user is allowed to edit the text in this control. (Inherited from SparkSkinnableTextBase)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)

Name	Description
enableIME	A flag that indicates whether the IME should be enabled when the component receives focus. (Inherited from SparkSkinnableTextBase)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from SparkSkinnableTextBase)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
heightInLines	The default height of the control, measured in lines.
id	the ID of the component. (Inherited from FlexObject)
imeMode	the IME (input method editor) mode. (Inherited from SparkSkinnableTextBase)
lastVisibleRow	the index of the last visible child. (Inherited from SparkSkinnableTextBase)
maxChars	The maximum number of characters that the text field can contain, as entered by a user. (Inherited from SparkSkinnableTextBase)
maxWidth	The maximum recommended width of the component to be considered by the parent during layout. (Inherited from SparkSkinnableTextBase)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available. (Inherited from SparkSkinnableTextBase)
numRows	the total number of rows of data available. (Inherited from SparkSkinnableTextBase)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)

Name	Description
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
restrict	the set of characters that a user can enter into the text field. (Inherited from SparkSkinnableTextBase)
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selectable	A flag indicating whether the content is selectable with the mouse, or with the keyboard when the control has the keyboard focus. (Inherited from SparkSkinnableTextBase)
selectionActivePosition	A character position, relative to the beginning of the text String, specifying the end of the selection that moves when the selection is extended with the arrow keys. (Inherited from SparkSkinnableTextBase)
selectionAnchorPosition	A character position, relative to the beginning of the text String, specifying the end of the selection that stays fixed when the selection is extended with the arrow keys. (Inherited from SparkSkinnableTextBase)
selectionHighlighting	a value whether the text selection is highlighted. (Inherited from SparkSkinnableTextBase)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
text	The text String displayed by this component. (Inherited from SparkSkinnableTextBase)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>widthInChars</i>	The default width of the control, measured in em units.
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>ChangeFocus</i>	Changes the current focus. (Inherited from <i>FlexObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>DragCancel</i>	The event triggered when a drag operation is cancelled.
<i>DragDrop</i>	The event triggered when the dragged item is dropped.
<i>DragStart</i>	The event triggered when the drag/drop operation begins.
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableTextBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Input	Dispatched when the user types, deletes, or pastes text into the control. (Inherited from SparkSkinnableTextBase)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when the mouse is used to move the scroll bars on this container. (Inherited from SparkSkinnableTextBase)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SelectText	Dispatched when text is selected. (Inherited from SparkSkinnableTextBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkSkinnableTextBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkTextBase Class

Description

The base class for Spark text controls such as `Label` and `RichText` which display text using CSS styles for the default format.

Inheritance Hierarchy

- [FlexObject](#)
 - `SparkTextBase`
 - [SparkLabel](#)

- [SparkRichText](#)

Syntax

```
'Declaration
Public Class SparkTextBase _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
isTruncated	whether the text has been truncated.
maxDisplayedLines	An integer which determines whether, and where, the text gets truncated.
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)

Name	Description
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>text</i>	The text displayed by this text component.
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	The event triggered when a drag operation is cancelled.
DragDrop	The event triggered when the dragged item is dropped.
DragStart	The event triggered when the drag/drop operation begins.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter,

Name	Description
WaitForScreenshotStable	<p>the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

SparkTextInput Class

Description

TextInput is a text-entry control that lets users enter and edit a single line of uniformly-formatted text.

Inheritance Hierarchy

- [SparkSkinnableTextBase](#)
 - SparkTextInput

Syntax

```
'Declaration
Public Class SparkTextInput _
Inherits SparkSkinnableTextBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableTextBase)
currentState	the current view state of the component. (Inherited from FlexObject)
displayAsPassword	whether the text field is a password text field. (Inherited from SparkSkinnableTextBase)

Name	Description
<i>editable</i>	A flag indicating whether the user is allowed to edit the text in this control. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>enableIME</i>	A flag that indicates whether the IME should be enabled when the component receives focus. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>imeMode</i>	the IME (input method editor) mode. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>maxChars</i>	The maximum number of characters that the text field can contain, as entered by a user. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>maxWidth</i>	The maximum recommended width of the component to be considered by the parent during layout. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableTextBase</i>)

Name	Description
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>restrict</i>	the set of characters that a user can enter into the text field. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selectable</i>	A flag indicating whether the content is selectable with the mouse, or with the keyboard when the control has the keyboard focus. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>selectionActivePosition</i>	A character position, relative to the beginning of the text String, specifying the end of the selection that moves when the selection is extended with the arrow keys. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>selectionAnchorPosition</i>	A character position, relative to the beginning of the text String, specifying the end of the selection that stays fixed when the selection is extended with the arrow keys. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>selectionHighlighting</i>	a value whether the text selection is highlighted. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>text</i>	The text String displayed by this component. (Inherited from <i>SparkSkinnableTextBase</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
widthInChars	The default width of the control, measured in em units.
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	The event triggered when a drag operation is cancelled.
DragDrop	The event triggered when the dragged item is dropped.
DragStart	The event triggered when the drag/drop operation begins.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableTextBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Input	Dispatched when the user types, deletes, or pastes text into the control. (Inherited from SparkSkinnableTextBase)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
MouseScroll	Dispatched when the mouse is used to move the scroll bars on this container. (Inherited from SparkSkinnableTextBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SelectText	Dispatched when text is selected. (Inherited from SparkSkinnableTextBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkSkinnableTextBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForProperty	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkTileGroup Class

Description

The TileGroup container is an instance of the Group container that uses the TileLayout class.

Inheritance Hierarchy

- [SparkGroup](#)
 - SparkTileGroup

Syntax

```
'Declaration  
Public Class SparkTileGroup _  
Inherits SparkGroup
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	a value whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkGroupBase)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
blendMode	A value from the BlendMode class that specifies which blend mode to use. (Inherited from SparkGroup)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
clipAndEnableScrolling	a value whether to clip the children to the boundaries of the viewport. (Inherited from SparkGroupBase)
columnNames	a list containing the names of all columns in the data. (Inherited from SparkGroupBase)
contentHeight	The height of the viewport's content. (Inherited from SparkGroupBase)
contentWidth	The width of the viewport's contents. (Inherited from SparkGroupBase)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from SparkGroupBase)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)

Name	Description
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
horizontalScrollPosition	The x coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component (Inherited from SparkGroupBase)
id	the ID of the component. (Inherited from FlexObject)
lastVisibleRow	the index of the last visible child. (Inherited from SparkGroupBase)
luminosityClip	a value whether the luminosity mask clips the masked content. (Inherited from SparkGroupBase)
luminosityInvert	a value that controls the calculation of the RGB color value of a graphic element being masked by a luminosity mask. (Inherited from SparkGroupBase)
maskType	The mask type. (Inherited from SparkGroupBase)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
mouseEnabledWhereTransparent	a value whether the entire bounds of the Group respond to mouse events such as click and roll over. (Inherited from SparkGroupBase)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)
numColumns	the total number of columns in the data available. (Inherited from SparkGroupBase)
numElements	The number of visual elements in this container. (Inherited from SparkGroupBase)
numRows	the total number of rows of data available. (Inherited from SparkGroupBase)
percentHeight	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
percentWidth	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
resizeMode	The ResizeMode for this container. (Inherited from SparkGroupBase)

Name	Description
<i>scaleGridBottom</i>	the bottom coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridLeft</i>	the left coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridRight</i>	the right coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleGridTop</i>	the top coordinate of the scale grid. (Inherited from <i>SparkGroup</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>verticalScrollPosition</i>	The y coordinate of the origin of the viewport in the component's coordinate system, where the default value is (0,0) corresponding to the upper-left corner of the component. (Inherited from <i>SparkGroupBase</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkGroupBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkGroupBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkGroupBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkGroupBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MouseScroll	Dispatched when the mouse is used to move the scroll bars on this container. (Inherited from SparkGroupBase)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Scroll	Dispatched when the container's scroll events occur. This event is dispatched if the scrolling is done programmatically or by the user. (Inherited from SparkGroupBase)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkGroupBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkTitleWindow Class

Description

The `TitleWindow` class extends `Panel` to include a close button and move area.

Inheritance Hierarchy

- [SparkPanel](#)
 - SparkTitleWindow

Syntax

```
'Declaration
Public Class SparkTitleWindow _
Inherits SparkPanel
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoLayout	whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)

Name	Description
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>controlBarVisible</i>	a value whether the control bar is visible. (Inherited from <i>SparkPanel</i>)
<i>creationPolicy</i>	the content creation policy for this component. (Inherited from <i>SparkSkinnableContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>deferredContentCreated</i>	whether deferred content has been created. (Inherited from <i>SparkSkinnableContainer</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)

Name	Description
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkSkinnableContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>title</i>	the title or caption displayed in the title bar. (Inherited from <i>SparkPanel</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>GetStyle</i>	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
<i>GetValues</i>	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>Move</i>	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkPanel)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> ,

Name	Description
	because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkToggleButton Class

Description

The ToggleButton component defines a toggle button. Clicking the button toggles it between the up and an down states.

Inheritance Hierarchy

- [SparkToggleButtonBase](#)
 - SparkToggleButton
 - [SparkButtonBarButton](#)

Syntax

```
'Declaration
Public Class SparkToggleButton _
Inherits SparkToggleButtonBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)

Name	Description
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>autoRepeat</i>	whether to dispatch repeated buttonDown events if the user holds down the mouse button. (Inherited from <i>SparkButtonBase</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text to appear on the ButtonBase control. (Inherited from <i>SparkButtonBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)

Name	Description
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
selected	whether the button is in the down state, and false if it is in the up state. (Inherited from SparkToggleButtonBase)
stickyHighlighting	a value whether the button displays its down skin when the user presses it but changes to its over skin when the user drags the mouse off of it. (Inherited from SparkButtonBase)
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TextObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TextObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	<p>parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	The event triggered when a drag operation is cancelled.
DragDrop	The event triggered when the dragged item is dropped.
DragStart	The event triggered when the drag/drop operation begins.
Exists	<p>Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)</p>
Exists	<p>Checks if an object exists in the application under test. (Inherited from TestObject)</p>
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)

Name	Description
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkButtonBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkToggleButtonBase Class

Description

The `ToggleButtonBase` component is the base class for the Spark button components that support the selected property.

Inheritance Hierarchy

- [SparkButtonBase](#)
 - SparkToggleButtonBase
 - [SparkCheckBox](#)
 - [SparkRadioButton](#)
 - [SparkToggleButton](#)

Syntax

```
'Declaration
Public Class SparkToggleButtonBase _
Inherits SparkButtonBase
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
autoRepeat	whether to dispatch repeated <code>buttonDown</code> events if the user holds down the mouse button. (Inherited from SparkButtonBase)
className	the name of this instance's class, such as <code>Button</code> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)

Name	Description
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>label</i>	the text to appear on the ButtonBase control. (Inherited from <i>SparkButtonBase</i>)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>selected</i>	whether the button is in the down state, and false if it is in the up state.
<i>stickyHighlighting</i>	a value whether the button displays its down skin when the user presses it but changes to its over skin when the user drags the mouse off of it. (Inherited from <i>SparkButtonBase</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)

Name	Description
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkButtonBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkTrackBase Class

Description

The TrackBase class is a base class for components with a track and one or more thumb buttons, such as Slider and ScrollBar.

Inheritance Hierarchy

- [SparkRange](#)
 - SparkTrackBase
 - [SparkScrollBar](#)
 - [SparkSlider](#)

Syntax

```
'Declaration  
Public Class SparkTrackBase _  
Inherits SparkRange
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
maximum	The maximum valid value. (Inherited from SparkRange)
minimum	The minimum valid value. (Inherited from SparkRange)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
numAutomationChildren	the number of automation children this container has. (Inherited from FlexObject)

Name	Description
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>snapInterval</i>	the snapInterval. (Inherited from <i>SparkRange</i>)
<i>stepSize</i>	The amount that the value property changes when the changeValueByStep() method is called. (Inherited from <i>SparkRange</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>value</i>	The current value for this range. (Inherited from <i>SparkRange</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when the value of the NumericStepper changes as a result of user interaction.
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkVideoDisplay Class

Description

Identifies a Spark video display control.

Inheritance Hierarchy

- [FlexObject](#)
 - SparkVideoDisplay

Syntax

```
'Declaration
Public Class SparkVideoDisplay _
Inherits FlexObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
autoDisplayFirstFrame	whether the video is loaded when the source is set.
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
autoPlay	whether the video starts playing immediately when the source property is set.
autoRewind	whether the FLV file should rewind to the first frame when play stops.
bytesLoaded	the number of bytes of data that have been downloaded into the application.
bytesTotal	the total size in bytes of the data being downloaded into the application.
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)

Name	Description
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>currentTime</i>	the current time of the playhead, measured in seconds, since the video starting playing.
<i>duration</i>	the duration of the video's playback, in seconds.
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>loop</i>	whether the media should play again after playback has completed.
<i>mediaPlayerState</i>	the current state of the video.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>muted</i>	whether the video is muted.
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>pauseWhenHidden</i>	whether the video continues to play when it is "hidden".
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>playing</i>	a value whether the video is playing or is attempting to play.
<i>scaleMode</i>	the different ways of sizing the video content.

Name	Description
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>volume</i>	the volume level, specified as a value between 0 and 1.
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkVideoPlayer Class

Description

The `VideoPlayer` control is a skinnable video player that supports progressive download, multi-bitrate streaming, and streaming video.

Inheritance Hierarchy

- [SparkObject](#)
 - SparkVideoPlayer

Syntax

```
'Declaration  
Public Class SparkVideoPlayer _  
Inherits SparkObject
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)
height	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
id	the ID of the component. (Inherited from FlexObject)
mouseEnabled	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
muted	a value whether the control is muted.

Name	Description
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)
<i>playing</i>	a value if the video is playing or is attempting to play.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from <i>FlexObject</i>)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from <i>FlexObject</i>)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from <i>FlexDisplayObject</i>)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from <i>FlexDisplayObject</i>)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from <i>FlexDisplayObject</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>themeColor</i>	the theme color of a component. (Inherited from <i>FlexObject</i>)
<i>toolTip</i>	the text to display in the ToolTip. The default value is Nothing. (Inherited from <i>FlexObject</i>)
<i>useHandCursor</i>	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from <i>FlexObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>visible</i>	whether the display object is visible. (Inherited from <i>FlexDisplayObject</i>)
<i>volume</i>	The volume level, specified as a value between 0 and 1.
<i>width</i>	the width of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>x</i>	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)
<i>y</i>	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from <i>FlexDisplayObject</i>)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkVolumeBar Class

Description

The VolumeBar class defines a drop-down slider to control the volume of the VideoDisplay control.

Inheritance Hierarchy

- [SparkSlider](#)
 - SparkVolumeBar

Syntax

```
'Declaration
Public Class SparkVolumeBar _
Inherits SparkSlider
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
currentState	the current view state of the component. (Inherited from FlexObject)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)

Name	Description
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>isDropDownOpen</i>	a value whether the volumebar is open.
<i>maximum</i>	The maximum valid value. (Inherited from SparkRange)
<i>minimum</i>	The minimum valid value. (Inherited from SparkRange)
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from FlexDisplayObject)
<i>muted</i>	a value whether the control is muted.
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
<i>scaleY</i>	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
<i>snapInterval</i>	the snapInterval. (Inherited from SparkRange)
<i>stepSize</i>	The amount that the value property changes when the changeValueByStep() method is called. (Inherited from SparkRange)
<i>tabChildren</i>	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
<i>tabEnabled</i>	whether this object is in the tab order. (Inherited from FlexDisplayObject)
<i>tabIndex</i>	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
<i>Text</i>	The text of the control. (Inherited from TestObject)
<i>themeColor</i>	the theme color of a component. (Inherited from FlexObject)

Name	Description
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
value	The current value for this range. (Inherited from SparkRange)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Change	Dispatched when the value of the NumericStepper changes as a result of user interaction. (Inherited from SparkTrackBase)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)

Name	Description
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from FlexObject)
MuteChange	Dispatched when the mute status changes as a result of user interaction.
Open	Dispatched when the user clicks the drop-down button to display the drop-down list.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkTrackBase)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForProperty	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkWindow Class

Description

The Window is a top-level container for additional windows in an AIR desktop application.

Inheritance Hierarchy

- [SparkSkinnableContainer](#)
 - SparkWindow

Syntax

```
'Declaration  
Public Class SparkWindow _  
Inherits SparkSkinnableContainer _  
Implements IMoveable
```

Properties

Name	Description
alpha	the alpha transparency value of the object specified. (Inherited from FlexDisplayObject)
alwaysInFront	a value whether the underlying NativeWindow is always in front of other windows.
autoLayout	whether measurement and layout are done when the position or size of a child is changed. (Inherited from SparkSkinnableContainer)
automationClassName	a string containing the automation class name for the element. (Inherited from FlexObject)
automationIndex	a string containing the automation index for the element. (Inherited from FlexObject)
automationName	the name that can be used as an identifier for this object. (Inherited from FlexObject)
className	the name of this instance's class, such as <i>Button</i> . (Inherited from FlexObject)
closed	a value whether the underlying window has been closed.
columnNames	a list containing the names of all columns in the data. (Inherited from SparkSkinnableContainerBase)
creationPolicy	the content creation policy for this component. (Inherited from SparkSkinnableContainer)
currentState	the current view state of the component. (Inherited from FlexObject)
deferredContentCreated	whether deferred content has been created. (Inherited from SparkSkinnableContainer)
enabled	whether the component can accept user interaction. (Inherited from FlexObject)
errorColor	the color of the component highlight when validation fails. (Inherited from FlexObject)
errorString	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from FlexObject)
firstVisibleRow	the index of the first visible child. (Inherited from SparkSkinnableContainerBase)
focusEnabled	whether the component can receive focus when tabbed to. (Inherited from FlexObject)

Name	Description
<i>height</i>	the height of the display object, in pixels. (Inherited from FlexDisplayObject)
<i>id</i>	the ID of the component. (Inherited from FlexObject)
<i>isAIRWindow</i>	a value whether this window is an AIR window.
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from SparkSkinnableContainerBase)
<i>maxHeight</i>	The maximum recommended height of the component to be considered by the parent during layout.
<i>maximizable</i>	a value whether the window can be maximized.
<i>maxWidth</i>	The maximum recommended width of the component to be considered by the parent during layout.
<i>minHeight</i>	The minimum recommended height of the component to be considered by the parent during layout.
<i>minimizable</i>	a value whether the window can be minimized.
<i>minWidth</i>	The minimum recommended width of the component to be considered by the parent during layout.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use <code>flash.display.DisplayObjectContainer.mouseChildren</code> . (Inherited from FlexDisplayObject)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from FlexObject)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from SparkSkinnableContainerBase)
<i>numElements</i>	The number of visual elements in this container. (Inherited from SparkSkinnableContainer)
<i>numRows</i>	the total number of rows of data available. (Inherited from SparkSkinnableContainerBase)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from FlexObject)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from FlexObject)
<i>resizable</i>	a value whether the window can be resized.
<i>scaleX</i>	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)

Name	Description
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
showStatusBar	a value whether the status bar is visible.
status	The string that appears in the status bar, if it is visible.
systemChrome	the type of system chrome (if any) the window has.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
title	The title that appears in the window title bar and the taskbar.
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
transparent	a value whether the window is transparent.
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)

Name	Description
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinnableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from IMoveable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StateChange	Dispatched when the displaystatus of the window is changed.
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SparkWindowedApplication Class

Description

The `WindowedApplication` defines the application container that you use to create Flex applications for AIR applications.

Inheritance Hierarchy

- [SparkApplication](#)
 - `SparkWindowedApplication`

Syntax

'Declaration

```
Public Class SparkWindowedApplication _
```

Inherits SparkApplication _
Implements IMoveable

Properties

Name	Description
<i>alpha</i>	the alpha transparency value of the object specified. (Inherited from <i>FlexDisplayObject</i>)
<i>alwaysInFront</i>	a value whether the underlying NativeWindow is always in front of other windows.
<i>applicationID</i>	The identifier that AIR uses to identify the application.
<i>autoExit</i>	a value whether the AIR application will quit when the last window closes or will continue running in the background.
<i>autoLayout</i>	whether measurement and layout are done when the position or size of a child is changed. (Inherited from <i>SparkSkinnableContainer</i>)
<i>automationClassName</i>	a string containing the automation class name for the element. (Inherited from <i>FlexObject</i>)
<i>automationIndex</i>	a string containing the automation index for the element. (Inherited from <i>FlexObject</i>)
<i>automationName</i>	the name that can be used as an identifier for this object. (Inherited from <i>FlexObject</i>)
<i>className</i>	the name of this instance's class, such as <i>Button</i> . (Inherited from <i>FlexObject</i>)
<i>closed</i>	a value whether the underlying window has been closed.
<i>colorCorrection</i>	The value of the stage's colorCorrection property. (Inherited from <i>SparkApplication</i>)
<i>columnNames</i>	a list containing the names of all columns in the data. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>controlBarVisible</i>	whether the control bar is visible. (Inherited from <i>SparkApplication</i>)
<i>creationPolicy</i>	the content creation policy for this component. (Inherited from <i>SparkSkinnableContainer</i>)
<i>currentState</i>	the current view state of the component. (Inherited from <i>FlexObject</i>)
<i>deferredContentCreated</i>	whether deferred content has been created. (Inherited from <i>SparkSkinnableContainer</i>)
<i>enabled</i>	whether the component can accept user interaction. (Inherited from <i>FlexObject</i>)
<i>errorColor</i>	the color of the component highlight when validation fails. (Inherited from <i>FlexObject</i>)
<i>errorString</i>	the text that will be displayed by a component's error tip when a component is monitored and validation fails. (Inherited from <i>FlexObject</i>)

Name	Description
<i>firstVisibleRow</i>	the index of the first visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>focusEnabled</i>	whether the component can receive focus when tabbed to. (Inherited from <i>FlexObject</i>)
<i>height</i>	the height of the display object, in pixels. (Inherited from <i>FlexDisplayObject</i>)
<i>id</i>	the ID of the component. (Inherited from <i>FlexObject</i>)
<i>lastVisibleRow</i>	the index of the last visible child. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>maxHeight</i>	The maximum recommended height of the component to be considered by the parent during layout.
<i>maximizable</i>	a value whether the window can be maximized.
<i>maxWidth</i>	The maximum recommended width of the component to be considered by the parent during layout.
<i>minHeight</i>	The minimum recommended height of the component to be considered by the parent during layout.
<i>minimizable</i>	a value whether the window can be minimized.
<i>minWidth</i>	The minimum recommended width of the component to be considered by the parent during layout.
<i>mouseEnabled</i>	whether this object receives mouse messages. The default value is true, which means that by default any InteractiveObject instance that is on the display list receives mouse events. If mouseEnabled is set to false, the instance does not receive any mouse events. Any children of this instance on the display list are not affected. To change the mouseEnabled behavior for all children of an object on the display list, use flash.display.DisplayObjectContainer.mouseChildren. (Inherited from <i>FlexDisplayObject</i>)
<i>numAutomationChildren</i>	the number of automation children this container has. (Inherited from <i>FlexObject</i>)
<i>numColumns</i>	the total number of columns in the data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>numElements</i>	The number of visual elements in this container. (Inherited from <i>SparkSkinnableContainer</i>)
<i>numRows</i>	the total number of rows of data available. (Inherited from <i>SparkSkinnableContainerBase</i>)
<i>percentHeight</i>	the number that specifies the height of a component as a percentage of its parent's size. (Inherited from <i>FlexObject</i>)
<i>percentWidth</i>	the number that specifies the width of a component as a percentage of its parent's size. Allowed values are 0-100. The default value is NaN. Setting the width or explicitWidth properties resets this property to NaN. (Inherited from <i>FlexObject</i>)

Name	Description
resizable	a value whether the window can be resized.
scaleX	the number that specifies the horizontal scaling factor. (Inherited from FlexObject)
scaleY	the number that specifies the vertical scaling factor. (Inherited from FlexObject)
showStatusBar	a value whether the status bar is visible.
status	The string that appears in the status bar, if it is visible.
systemChrome	the type of system chrome (if any) the window has.
tabChildren	whether the children of the display object are tab enabled. The default is true. (Inherited from FlexDisplayObject)
tabEnabled	whether this object is in the tab order. (Inherited from FlexDisplayObject)
tabIndex	the tab ordering of objects in a SWF file. (Inherited from FlexDisplayObject)
Text	The text of the control. (Inherited from TestObject)
themeColor	the theme color of a component. (Inherited from FlexObject)
title	The title that appears in the window title bar and the taskbar.
toolTip	the text to display in the ToolTip. The default value is Nothing. (Inherited from FlexObject)
transparent	a value whether the window is transparent.
url	The URL from which this Application's SWF file was loaded. (Inherited from SparkApplication)
useHandCursor	whether the pointing hand cursor appears when the mouse rolls over an element. The default value is true. If useHandCursor is false, the arrow pointer is used instead. (Inherited from FlexObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
viewSourceURL	the URL where the application's source can be viewed. (Inherited from SparkApplication)
visible	whether the display object is visible. (Inherited from FlexDisplayObject)
width	the width of the display object, in pixels. (Inherited from FlexDisplayObject)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)
x	the x coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Name	Description
y	the y coordinate of the DisplayObject instance relative to the local coordinates of the parent DisplayObjectContainer. (Inherited from FlexDisplayObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeFocus	Changes the current focus. (Inherited from FlexObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DragCancel	Dispatched when a drag operation is cancelled. (Inherited from SparkSkinnableContainerBase)
DragDrop	Dispatched when the dragged item is dropped. (Inherited from SparkSkinnableContainerBase)
DragStart	Dispatched when the drag/drop operation begins. (Inherited from SparkSkinnableContainerBase)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyle	Returns the style value for the given style. For available styles consider the Apache Flex sdk documentation. (Inherited from FlexDisplayObject)
GetValues	Returns a matrix containing the automation values of all parts of the components. (Inherited from SparkSkinableContainerBase)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the component to a specified position within its parent. Calling this method is the same as setting the component's x and y properties. (Inherited from IMoveable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Sets the focus to this component. The component may in turn pass focus to a subcomponent. (Inherited from FlexObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ToString	Returns a string representation of the control. (Inherited from FlexDisplayObject)
Type	Dispatched when the user presses a key. (Inherited from SparkApplication)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Java SWT Class Reference

Lists the available classes for testing Java SWT controls.

CBanner Class

Description

The class used to layout the toolbar area and perspective switching toolbar in the workbench.

Inheritance Hierarchy

- [Control](#)
 - CBanner

Syntax

```
'Declaration  
Public Class CBanner _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Bottom	The control located at the bottom of the banner.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
Left	The control located on the left side of the banner.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Right	The control located on the right side of the banner.
RightMinimumSize	The minimum size of the control located on the right side of the banner.
RightWidth	The width of the control located on the right side of the banner.
Simple	Whether the banner is rendered with a simple, traditional shape.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

CoolBar Class

Description

A container class that contains SWT widgets, primarily toolbars.

Inheritance Hierarchy

- [Control](#)
 - CoolBar

Syntax

```
'Declaration
Public Class CoolBar _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the toolbar.
Items	A list of items in the toolbar.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need

Name	Description
	this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

CoolItem Class

Description

Contains selectable user interface objects that represent the areas of a CoolBar that can be dynamically positioned.

Inheritance Hierarchy

- [Item](#)
 - CoolItem

Syntax

```
'Declaration
Public Class CoolItem _
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Content	The content of the areas that can be dynamically positioned.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

CtabFolder Class

Description

Class for an enhanced tab folder, typically having decorated tabs and the minimize, maximize, and close buttons.

Inheritance Hierarchy

- [Control](#)
 - CTabFolder

Syntax

```
'Declaration  
Public Class CTabFolder _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the tab folder.
Items	A list of items in the tab folder.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	The index of the selected item.
SelectedItem	The selected tab item.
SelectedItemText	The text of the selected item or an empty string if no item is selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the visible page.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this <code>TestObject</code> . (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>Maximize</i>	Resizes the CTabFolder to its full (maximized) size.
<i>Minimize</i>	Resizes the CTabFolder to the minimum size allowed.
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Restore</i>	Restores the CTabFolder to its previous size.
<i>Select</i>	Selects an item from the CTabFolder.
<i>SelectList</i>	Selects one or more items in the tab.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

CTabItem Class

Description

The class for an enhanced tab within a CTabFolder.

Inheritance Hierarchy

- [Item](#)
 - CTabItem

Syntax

```
'Declaration
Public Class CTabItem _
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Selected	Whether the item is selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the visible tab.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects a tab.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ExpandBar Class

Description

The class for widgets that expand and close by clicking on a header.

Inheritance Hierarchy

- [Control](#)
 - ExpandBar

Syntax

```
'Declaration  
Public Class ExpandBar _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
ExpandedItems	A list of the items that are expanded.
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the expandbar.
Items	A list of items in the expandbar.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default

Name	Description
	<p>bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
Collapse	Collapses the ExpandBar.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Expand	Expands an ExpandBar.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ExpandItem Class

Description

The class that contains the expandable items in an ExpandBar.

Inheritance Hierarchy

- [Item](#)
 - ExpandItem

Syntax

```
'Declaration
Public Class ExpandItem _
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Content	The content of the item.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Expanded	Whether the item is expanded.
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Collapse	Collapses the expandable items.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Expand	Expands the expandable items.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)

Name	Description
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

HorizontalSash Class

Description

The class for the horizontal border around a window pane that can be dragged to adjust the window size.

Inheritance Hierarchy

- [Sash](#)
 - HorizontalSash

Syntax

```
'Declaration  
Public Class HorizontalSash _  
Inherits Sash
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel

Name	Description
	must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window pane that the sash borders. (Inherited from Sash)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Sash Class

Description

The class for the border around a window pane that can be dragged to adjust the window size.

Inheritance Hierarchy

- [Control](#)
 - Sash
 - [HorizontalSash](#)
 - [VerticalSash](#)

Syntax

```
'Declaration  
Public Class Sash _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The

Name	Description
	<p>captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Move</i>	Moves the window pane that the sash borders.
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SashForm Class

Description

The class for the border around a form that can be dragged to adjust the size of the form.

Inheritance Hierarchy

- [Control](#)
 - SashForm

Syntax

```
'Declaration  
Public Class SashForm _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
Horizontal	Whether the sash form is horizontal.
MaximizedControl	The sash window that is maximized in the form.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Vertical	Whether the sash form is vertical.
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
Weights	The relative sizes of the children of the sash form window.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ScrollableControl Class

Description

The implementation class for all controls which are only classified as controls but also contain a scrollbar.

Inheritance Hierarchy

- [Control](#)
 - ScrollableControl

Syntax

```
'Declaration
Public Class ScrollableControl _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default

Name	Description
	<p>bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ScrolledComposite Class

Description

The class for scrollbars and scrolls the content when the user uses the scrollbars.

Inheritance Hierarchy

- [Control](#)
 - ScrolledComposite

Syntax

```
'Declaration  
Public Class ScrolledComposite _  
Inherits Control
```

Properties

Name	Description
AlwaysShowScrollBars	Whether the scrollbar shows.
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Content	The content of the areas that can be dynamically positioned.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Origin	The point in the content that currently appears in the top left corner of the scrolled composite.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Shell Class

Description

The class for Java SWT windows.

Inheritance Hierarchy

- [Window](#)
 - Shell

Syntax

```
'Declaration
Public Class Shell _
Inherits Window
```

Properties

Name	Description
Application	The name of the Application that this Window belongs to. (Inherited from Window)
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this <code>TestObject</code> . (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT

Name	Description
	takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Spinner Class

Description

The class to enter and modify numeric values for up/down controls.

Inheritance Hierarchy

- [Control](#)
 - Spinner

Syntax

```
'Declaration
Public Class Spinner _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)

Name	Description
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
IncrementCount	The amount to increase the position value of the spinner control after pressing it.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	The internal page size of the spinner control.
Position	The position for the up/down control.
Range	The range of the spinner control.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Decrement	Decreases the value when the spinner controls are pressed.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetHorizontalBar</i>	Returns the horizontal scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Increment</i>	Increases the value when the spinner controls are pressed.
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)

Name	Description
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PageDecrement</i>	Decreases the value when the page spinner controls are pressed.
<i>PageIncrement</i>	Increases the value when the page spinner controls are pressed.
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>SetValue</i>	Sets the value for the spinner control.
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

StyledText Class

Description

The class for an editable user interface object that displays lines of text.

Inheritance Hierarchy

- [TextField](#)
 - `StyledText`

Syntax

```
'Declaration  
Public Class StyledText _  
Inherits TextField
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Editable	Whether the text is editable.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
IsPassword	Whether the control is a password text field. (Inherited from TextField)
MultiLine	Whether the control is multiline. (Inherited from TextField)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Position	The position of the caret within the text field. (Inherited from TextField)
SelectedRange	The selected range within the text field. (Inherited from TextField)
SelectedText	The selected text within the text field. An empty string if no text is selected. (Inherited from TextField)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel

Name	Description
	must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ClearText	Removes all text from the text field. (Inherited from TextField)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetMultiText	Returns the specified lines of text in the multi-line text field. (Inherited from TextField)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetMultiText	Substitutes all or part of the lines in the multi-line text field. (Inherited from TextField)
SetPosition	Sets the insertion point in the text field. (Inherited from TextField)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the specified range of the single-line or multi-line text field. (Inherited from TextField)
SetText	Substitutes new text for all or part of the text in the text field. (Inherited from TextField)
StartScreenRecording	Starts screen recording (Inherited from TestObject)

Name	Description
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SWTBrowser Class

Description

The class for the browser widget.

Inheritance Hierarchy

- [Control](#)
 - SWTBrowser

Syntax

```
'Declaration
Public Class SWTBrowser _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SWTDateTime Class

Description

The class for SWT date and time elements in a calendar, date, or time control.

Inheritance Hierarchy

- [Control](#)
 - SWTDateTime

Syntax

```
'Declaration
Public Class SWTDateTime _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
CurrentDate	The current date.
Day	The day.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
Hours	The hour.
Minutes	The minute.
Month	The month.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Seconds	The second.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application

Name	Description
	under test gets invisible during replay. (Inherited from GuiTestObject)
Year	The year.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects a date and time.
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForPropertyNotEquals 	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SWTTabControl Class

Description

The class for SWT TabFolder.

Inheritance Hierarchy

- [TabControl](#)
 - SWTTabControl

Syntax

```
'Declaration
Public Class SWTTabControl _
Inherits TabControl
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the tab control. (Inherited from TabControl)
Items	A list of items in the tab control. (Inherited from TabControl)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	The index of the selected item. (Inherited from TabControl)
SelectedItem	The selected item. (Inherited from TabControl)

Name	Description
SelectedTabItem	The name of the selected item.
TabItems	A list of TabItems in the tab control.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Sets the current page to the specified page. (Inherited from TabControl)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SWTTabItem Class

Description

The class for SWT TabItem.

Inheritance Hierarchy

- [Item](#)
 - SWTTabItem

Syntax

```
'Declaration
Public Class SWTTabItem _
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Selected	Whether the item is selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects a tab.
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForPropertyNotEquals 	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SWTTable Class

Description

The class for SWT tables.

Inheritance Hierarchy

- [Table](#)
 - SWTTable

Syntax

```
'Declaration
Public Class SWTTable _
Inherits Table
```

Properties

Name	Description
AllowsCheck	Whether the control can display a checkmark. (Inherited from Table)
AllowsMultiSelect	Whether the table supports selecting multiple items. (Inherited from Table)
Background	The background color of the GUI object. (Inherited from GuiTestObject)
ColumnCount	The number of columns in the table. (Inherited from Table)
ColumnItems	A list of all items in the column. (Inherited from Table)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
RowCount	The number of rows in the table. (Inherited from Table)
RowItems	A list of all table rows in the table. (Inherited from Table)
SelectedIndices	The indices of the selected item(s). (Inherited from Table)
SelectedItems	The selected item(s). (Inherited from Table)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box in the defined row of a table. (Inherited from Table)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleSelect	Double-clicks an item. (Inherited from Table)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
ExtendSelectRow	Selects a range of rows. (Inherited from Table)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FocusRow</i>	Focuses on a row in the table. (Inherited from <i>Table</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetHorizontalBar</i>	Returns the horizontal scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MultiSelectRow	Adds a row from the table to the set of selected rows. (Inherited from Table)
MultiUnselectRow	Removes a row from the set of selected rows. (Inherited from Table)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SelectRow	Selects an row in the table. (Inherited from Table)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks the check box in the defined row of a table. (Inherited from Table)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForProperty	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SWTTableColumn Class

Description

The class for columns in an SWT table.

Inheritance Hierarchy

- [TableColumn](#)
 - SWTTableColumn

Syntax

```
'Declaration  
Public Class SWTTableColumn _  
Inherits TableColumn
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
Width	The width of the column. (Inherited from TableColumn)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Resize	Resizes a column. (Inherited from TableColumn)
Select	Selects a column. (Inherited from TableColumn)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SWTTableRow Class

Description

The class for rows in an SWT table.

Inheritance Hierarchy

- [TableRow](#)
 - SWTTableRow

Syntax

```
'Declaration
Public Class SWTTableRow _
Inherits TableRow
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Checked	Whether the check box in the row is checked. (Inherited from TableRow)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the row. (Inherited from TableRow)
Items	A list of items in the row. (Inherited from TableRow)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> ,

Name	Description
WaitForObject	because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SWTTree Class

Description

The class for SWT tree widgets.

Inheritance Hierarchy

- [Tree](#)
 - SWTTree

Syntax

```
'Declaration
Public Class SWTTree _
Inherits Tree
```

Properties

Name	Description
AllowsCheck	Whether the control can display a checkmark. (Inherited from Tree)
AllowsMultiSelect	Whether the control supports selecting multiple items. (Inherited from Tree)
Background	The background color of the GUI object. (Inherited from GuiTestObject)

Name	Description
ColumnCount	The number of columns in the tree.
Columns	A list of columns in the tree.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the tree (including all children). (Inherited from Tree)
ItemPaths	A list of items in the tree (including all children). (Inherited from Tree)
Items	A list of items in the tree (including all children). (Inherited from Tree)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	The index of the first selected item. (Inherited from Tree)
SelectedIndices	The indices of the selected item(s). (Inherited from Tree)
SelectedItem	The name of the first selected item. (Inherited from Tree)
SelectedItems	The names of the selected item(s). (Inherited from Tree)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
VisibleItemPaths	A list of visible items in the tree (including all children). (Inherited from Tree)
VisibleItems	A list of visible items in the tree (including all children). (Inherited from Tree)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box. (Inherited from Tree)
Click	Clicks on the object. (Inherited from IClickable)
Collapse	Collapses an item in a treeview control. (Inherited from Tree)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Double-clicks an item. (Inherited from Tree)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Expand	Expands an object in a treeview control. (Inherited from Tree)
ExtendSelect	Selects a range of items by extending the selection.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetContentsOfColumn	Retrieves the contents of the specified column as a list. To retrieve the contents of any column greater than column one, you must specify the parameters <i>rawMode</i> and <i>column</i> .
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetItemPaths	Returns all items of the tree as a list of <i>itempaths</i> (Inherited from Tree)
GetItemRect	Returns the size and position of an item relative to the treeview control. (Inherited from Tree)
GetItemsOfColumn	Retrieve items of the specified column from Tree
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetSelectedTextOfColumn	Retrieves selected text from the specified column
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsExpandable	Checks if the given item can be expanded. (Inherited from Tree)
IsExpanded	Checks if the given item is expanded. (Inherited from Tree)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MultiSelect	Adds an item from the tree to the set of selected items.
MultiUnselect	Removes an item in the tree from the set of selected items.
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item from the tree. (Inherited from Tree)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks the check box. (Inherited from Tree)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By

Name	Description
	default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SWTTreeColumn Class

Description

The class for columns in a Tree widget.

Inheritance Hierarchy

- [Item](#)
 - SWTTreeColumn

Syntax

```
'Declaration
Public Class SWTTreeColumn _
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)

Name	Description
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
Width	The width of the column.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Resize	Resizes a column of the tree.
Select	Selects a tree column.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

VerticalSash Class

Description

The class for the vertical border around a window pane that can be dragged to adjust the window size.

Inheritance Hierarchy

- [Sash](#)
 - VerticalSash

Syntax

```
'Declaration
Public Class VerticalSash _
Inherits Sash
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window pane that the sash borders. (Inherited from Sash)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ViewForm Class

Description

The class to lay out a view's label/menu/toolbar local bar in the workbench.

Inheritance Hierarchy

- [Control](#)
 - ViewForm

Syntax

```
'Declaration
Public Class ViewForm _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Content	The content of the areas that can be dynamically positioned.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
TopCenter	The window located in the top center of the pane.
TopLeft	The window located in the top left of the pane.
TopRight	The window located in the top right of the pane.
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetHorizontalBar</i>	Returns the horizontal scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

キーワード駆動テスト クラス リファレンス

キーワード駆動テストで利用可能なクラスを一覧します。

ArgumentAttribute クラス

説明

キーワード駆動テストのキーワードとしてマークされるメソッドの引数の名前を明示的に指定するために使用されます。

構文

```
<AttributeUsageAttribute(AttributeTargets.Parameter)>
Public Class ArgumentAttribute Inherits Attribute
```

プロパティ

名前	説明
Name プロパティ (ArgumentAttribute)	引数の名前を取得します。

例 : 2つの引数を持つキーワードの作成

引数 username および password を持つ Login キーワードを作成するには、次のサンプルコードを使用します。

```
<Keyword()>
Public Sub Login(<Argument("Name of the user")> username As String,
<Argument("Password of the user")> password As String)
```

```
' keyword implementation
End Sub
```

例：説明のない引数の使用

引数の説明を使用せずに、引数 `username` および `password` を持つ `Login` キーワードを作成するには、次のサンプルコードを使用します。

```
<Keyword()>
Public Sub Login(username As String, password As String)
    ' keyword implementation
End Sub
```

```
<Keyword()>
Public Sub Login(<Argument> username As String, <Argument>
password As String)
    ' keyword implementation
End Sub
```

KeywordAttribute クラス

説明

メソッドがキーワードであることを指定します。キーワードの名前が指定されていない場合、アノテーションメソッドの名前がキーワード名として使用されます。

構文

```
'Declaration
<AttributeUsageAttribute(AttributeTargets.Method)>
Public Class KeywordAttribute Inherits Attribute
```

プロパティ

名前	説明
Name プロパティ (KeywordAttribute)	キーワードの名前を取得します。デフォルト値は空の文字列です。
Description プロパティ (KeywordAttribute)	キーワードの説明を取得します。デフォルト値は空の文字列です。

例：2つの引数を持つキーワードの作成

引数 `username` および `password` を持つ `Login` キーワードを作成するには、次のサンプルコードを使用します。

```
<Keyword()>
Public Sub Login(username As String, password As String)
    ' keyword implementation
End Sub
```

例：キーワードの名前の指定

新しいキーワードを作成するときにキーワードの名前を指定するには、次のサンプルコードを使用します。

```
<Keyword("Login user")>
Public Sub Login(username As String, password As String)
```

```
' keyword implementation
End Sub
```

KeywordGroupAttribute クラス

説明

クラスがキーワードグループであることを指定します。キーワードとしてマークされるこのクラス内のメソッドはキーワードです。これらのメソッドの完全修飾名は、キーワードグループの名前とキーワードの名前で構成されます。キーワードグループの名前がない場合、宣言しているクラスの単純名が代わりに使用されます。

構文

```
'Declaration
<AttributeUsageAttribute(AttributeTargets.Class)>
Public Class KeywordGroupAttribute Inherits Attribute
```

プロパティ

名前	説明
Name プロパティ (KeywordGroupAttribute)	キーワードグループの名前を取得します。デフォルト値は空の文字列です。

例：キーワードグループの作成

キーワードグループ *LoginPage* を作成するには、次のサンプルコードを使用します。

```
<KeywordGroup()>
Public Class LoginPage
    ' keyword methods
End Class
```

例：キーワードグループの名前の指定

引数の説明を使用せずに、引数 *username* および *password* を持つ *Login* キーワードを作成するには、次のサンプルコードを使用します。

```
<KeywordGroup("Login Page")>
Public Class LoginPage
    ' keyword methods
End Class
```

Mobile Class Reference

Lists the available classes for testing Mobile controls.

IMobileClickable Interface

Description

Interface for objects that use clicks.

Syntax

```
'Declaration  
Public Class IMobileClickable
```

Methods

Name	Description
Click	Clicks on the device at the specified coordinates.
DoubleClick	Double-clicks on the device at the specified coordinates.
LongClick	Long-clicks on the device at the specified coordinates.

IMobileGestures Interface

Description

Interface for objects that use long swipes.

Syntax

```
'Declaration  
Public Class IMobileGestures
```

Methods

Name	Description
Drag	Performs a drag operation between the two specified points.
MultiTouch	Performs a multi-touch operation with the given pointers. For every given pointer a swipe operation is executed at the same time.
PinchIn	Performs a two-pointer gesture, where each pointer moves toward the other, from the edges to the center of this object. For example to zoom out on an image.
PinchOut	Performs a two-pointer gesture, where each pointer moves opposite across the other, from the center out towards the edges of the this object. For example to zoom in on an image.
Swipe	Performs a swipe between the two specified points.
SwipeDown	Performs a down-swipe.
SwipeLeft	Performs a left-swipe.
SwipeRight	Performs a right-swipe.
SwipeUp	Performs a up-swipe.
SwipeWithSegments	Performs a swipe between the specified points.
TwoPointerMultiTouch	Performs a multi-touch operation with two pointers. It executes two swipe operation at the same time.

IMobileKeyable Interface

Description

Interface for objects that use keystrokes.

Syntax

```
'Declaration  
Public Class IMobileKeyable
```

Methods

Name	Description
TypeKeys	Sends a set of keystrokes to the object.

MobileButton Class

Description

Represents a mobile button. Mobile buttons can be clicked.

Inheritance Hierarchy

- [MobileObject](#)
 - MobileButton

Syntax

```
'Declaration  
Public Class MobileButton _  
Inherits MobileObject
```

Properties

Name	Description
IsEnabled	Whether the mobile object is enabled. (Inherited from MobileObject)
IsFocused	Whether the mobile object has focus. This property is not supported on iOS. (Inherited from MobileObject)
IsSelected	Whether the mobile object is selected or checked. (Inherited from MobileObject)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
DoubleClick	Double-clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
Drag	Performs a drag operation between the two specified points. (Inherited from IMobileGestures)
DragTo	Drags this object to the specified point relative to the mobile device. (Inherited from MobileObject)
DragToObject	Drags this object to the specified destination object. (Inherited from MobileObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>LongClick</i>	Long-clicks on the device at the specified coordinates. (Inherited from <i>IMobileClickable</i>)
<i>MultiTouch</i>	Performs a multi-touch operation with the given pointers. For every given pointer a swipe operation is executed at the same time. (Inherited from <i>IMobileGestures</i>)
<i>PinchIn</i>	Performs a two-pointer gesture, where each pointer moves toward the other, from the edges to the center of this object. For example to zoom out on an image. (Inherited from <i>IMobileGestures</i>)
<i>PinchOut</i>	Performs a two-pointer gesture, where each pointer moves opposite across the other, from the center out towards the edges of the this object. For example to zoom in on an image. (Inherited from <i>IMobileGestures</i>)
<i>ScrollIntoView</i>	Scroll until an object is visible. Works by dragging the scroll view in the indicated direction and then searching for the object, until either the object is found, or the repetition limit is reached (in which case an <i>ObjectNotFoundException</i> is thrown). (Inherited from <i>MobileObject</i>)
<i>SendImeAction</i>	Sends IME action to this object. (Inherited from <i>MobileObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>Swipe</i>	Performs a swipe between the two specified points. (Inherited from <i>IMobileGestures</i>)
<i>SwipeDown</i>	Performs a down-swipe. (Inherited from <i>IMobileGestures</i>)
<i>SwipeLeft</i>	Performs a left-swipe. (Inherited from <i>IMobileGestures</i>)
<i>SwipeRight</i>	Performs a right-swipe. (Inherited from <i>IMobileGestures</i>)
<i>SwipeUp</i>	Performs a up-swipe. (Inherited from <i>IMobileGestures</i>)

Name	Description
<i>SwipeWithSegments</i>	Performs a swipe between the specified points. (Inherited from <i>IMobileGestures</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TwoPointerMultiTouch</i>	Performs a multi-touch operation with two pointers. It executes two swipe operation at the same time. (Inherited from <i>IMobileGestures</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IMobileKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForObject</i>	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from <i>TestObject</i>)
<i>WaitForProperty</i>	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForPropertyNotEquals</i>	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

MobileDevice Class

Description

The mobile device.

Inheritance Hierarchy

- [TestObject](#)
 - MobileDevice

Syntax

```
'Declaration
Public Class MobileDevice _
Inherits TestObject _
Implements IMobileClickable, IMobileGestures, IMobileKeyable
```

Properties

Name	Description
DeviceId	The id of the device for the current connection.
DisplayHeight	The height of the display in pixels.
DisplayOrientation	The current orientation of the device.
DisplayWidth	The width of the display in pixels.
IsEmulator	Whether the connected device is an emulator.
Model	The device name of the connected device.
OperatingSystem	the operating system of the mobile device, Android or iOS
OsVersion	The version of the operating system of the connected device.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
CloseApp	Closes the app and releases the device, to make it available to other users.
DoubleClick	Double-clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
Drag	Performs a drag operation between the two specified points. (Inherited from IMobileGestures)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateConnectionString	Generates a connection string for this mobile device.
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>LongClick</i>	Long-clicks on the device at the specified coordinates. (Inherited from <i>IMobileClickable</i>)
<i>MultiTouch</i>	Performs a multi-touch operation with the given pointers. For every given pointer a swipe operation is executed at the same time. (Inherited from <i>IMobileGestures</i>)
<i>PinchIn</i>	Performs a two-pointer gesture, where each pointer moves toward the other, from the edges to the center of this object. For example to zoom out on an image. (Inherited from <i>IMobileGestures</i>)
<i>PinchOut</i>	Performs a two-pointer gesture, where each pointer moves opposite across the other, from the center out towards the edges of the this object. For example to zoom in on an image. (Inherited from <i>IMobileGestures</i>)
<i>PressBack</i>	Presses the back button. This method is not supported on iOS.
<i>PressDelete</i>	Presses the delete button.
<i>PressEnter</i>	Presses the enter button.
<i>PressHome</i>	Presses the home button.
<i>PressKeyCode</i>	Presses a certain Android-specific key code. This method is not supported on iOS.
<i>PressRecentApps</i>	Presses the recent apps button.
<i>Rotate</i>	Simulates rotating the device screen.

Name	Description
SetLocation	Sets the geographic location of the device, this method only works on Android devices (physical or emulated) and iOS simulators.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Sleep	Presses the power button if the screen is on. This method is not supported on iOS.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
Swipe	Performs a swipe between the two specified points. (Inherited from IMobileGestures)
SwipeDown	Performs a down-swipe. (Inherited from IMobileGestures)
SwipeLeft	Performs a left-swipe. (Inherited from IMobileGestures)
SwipeRight	Performs a right-swipe. (Inherited from IMobileGestures)
SwipeUp	Performs a up-swipe. (Inherited from IMobileGestures)
SwipeWithSegments	Performs a swipe between the specified points. (Inherited from IMobileGestures)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TwoPointerMultiTouch	Performs a multi-touch operation with two pointers. It executes two swipe operation at the same time. (Inherited from IMobileGestures)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IMobileKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException

Name	Description
	is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WakeUp	Presses the power button if the screen is off. This method is not supported on iOS.

MobileObject Class

Description

The base class for all mobile objects.

Inheritance Hierarchy

- [TestObject](#)
 - MobileObject
 - [MobileButton](#)
 - [MobileTextField](#)
 - [MobileWindow](#)

Syntax

'Declaration

```
Public Class MobileObject _
```

Inherits [TestObject](#) _
Implements [IMobileClickable](#), [IMobileGestures](#), [IMobileKeyable](#)

Properties

Name	Description
IsEnabled	Whether the mobile object is enabled.
IsFocused	Whether the mobile object has focus. This property is not supported on iOS.
IsSelected	Whether the mobile object is selected or checked.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
DoubleClick	Double-clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
Drag	Performs a drag operation between the two specified points. (Inherited from IMobileGestures)
DragTo	Drags this object to the specified point relative to the mobile device.
DragToObject	Drags this object to the specified destination object.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)

Name	Description
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
LongClick	Long-clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
MultiTouch	Performs a multi-touch operation with the given pointers. For every given pointer a swipe operation is executed at the same time. (Inherited from IMobileGestures)
PinchIn	Performs a two-pointer gesture, where each pointer moves toward the other, from the edges to the center of this object. For example to zoom out on an image. (Inherited from IMobileGestures)
PinchOut	Performs a two-pointer gesture, where each pointer moves opposite across the other, from the center out towards the

Name	Description
	edges of the this object. For example to zoom in on an image. (Inherited from IMobileGestures)
ScrollIntoView	Scroll until an object is visible. Works by dragging the scroll view in the indicated direction and then searching for the object, until either the object is found, or the repetition limit is reached (in which case an <code>ObjectNotFoundException</code> is thrown).
SendImeAction	Sends IME action to this object.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
Swipe	Performs a swipe between the two specified points. (Inherited from IMobileGestures)
SwipeDown	Performs a down-swipe. (Inherited from IMobileGestures)
SwipeLeft	Performs a left-swipe. (Inherited from IMobileGestures)
SwipeRight	Performs a right-swipe. (Inherited from IMobileGestures)
SwipeUp	Performs a up-swipe. (Inherited from IMobileGestures)
SwipeWithSegments	Performs a swipe between the specified points. (Inherited from IMobileGestures)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TwoPointerMultiTouch	Performs a multi-touch operation with two pointers. It executes two swipe operation at the same time. (Inherited from IMobileGestures)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IMobileKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

MobileTextField Class

Description

Represents a text field.

Inheritance Hierarchy

- [MobileObject](#)
 - MobileTextField

Syntax

```
'Declaration
Public Class MobileTextField _
Inherits MobileObject
```

Properties

Name	Description
IsEnabled	Whether the mobile object is enabled. (Inherited from MobileObject)
IsFocused	Whether the mobile object has focus. This property is not supported on iOS. (Inherited from MobileObject)
IsSelected	Whether the mobile object is selected or checked. (Inherited from MobileObject)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ClearText	Clears all the content from the text field.
Click	Clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
DoubleClick	Double-clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
Drag	Performs a drag operation between the two specified points. (Inherited from IMobileGestures)
DragTo	Drags this object to the specified point relative to the mobile device. (Inherited from MobileObject)
DragToObject	Drags this object to the specified destination object. (Inherited from MobileObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>LongClick</i>	Long-clicks on the device at the specified coordinates. (Inherited from <i>IMobileClickable</i>)
<i>MultiTouch</i>	Performs a multi-touch operation with the given pointers. For every given pointer a swipe operation is executed at the same time. (Inherited from <i>IMobileGestures</i>)
<i>PinchIn</i>	Performs a two-pointer gesture, where each pointer moves toward the other, from the edges to the center of this object. For example to zoom out on an image. (Inherited from <i>IMobileGestures</i>)

Name	Description
PinchOut	Performs a two-pointer gesture, where each pointer moves opposite across the other, from the center out towards the edges of the this object. For example to zoom in on an image. (Inherited from IMobileGestures)
ScrollIntoView	Scroll until an object is visible. Works by dragging the scroll view in the indicated direction and then searching for the object, until either the object is found, or the repetition limit is reached (in which case an <code>ObjectNotFoundException</code> is thrown). (Inherited from MobileObject)
SendImeAction	Sends IME action to this object. (Inherited from MobileObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetText	Substitutes new text for the text in the text field.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
Swipe	Performs a swipe between the two specified points. (Inherited from IMobileGestures)
SwipeDown	Performs a down-swipe. (Inherited from IMobileGestures)
SwipeLeft	Performs a left-swipe. (Inherited from IMobileGestures)
SwipeRight	Performs a right-swipe. (Inherited from IMobileGestures)
SwipeUp	Performs a up-swipe. (Inherited from IMobileGestures)
SwipeWithSegments	Performs a swipe between the specified points. (Inherited from IMobileGestures)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TwoPointerMultiTouch	Performs a multi-touch operation with two pointers. It executes two swipe operation at the same time. (Inherited from IMobileGestures)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IMobileKeyable)
TypePasswordKeys	Types an encrypted password into a text field.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

MobileWindow Class

Description

The mobile window.

Inheritance Hierarchy

- [MobileObject](#)
 - MobileWindow

Syntax

```
'Declaration  
Public Class MobileWindow _  
Inherits MobileObject
```

Properties

Name	Description
IsEnabled	Whether the mobile object is enabled. (Inherited from MobileObject)
IsFocused	Whether the mobile object has focus. This property is not supported on iOS. (Inherited from MobileObject)
IsSelected	Whether the mobile object is selected or checked. (Inherited from MobileObject)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
DoubleClick	Double-clicks on the device at the specified coordinates. (Inherited from IMobileClickable)
Drag	Performs a drag operation between the two specified points. (Inherited from IMobileGestures)
DragTo	Drags this object to the specified point relative to the mobile device. (Inherited from MobileObject)
DragToObject	Drags this object to the specified destination object. (Inherited from MobileObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>LongClick</i>	Long-clicks on the device at the specified coordinates. (Inherited from <i>IMobileClickable</i>)
<i>MultiTouch</i>	Performs a multi-touch operation with the given pointers. For every given pointer a swipe operation is executed at the same time. (Inherited from <i>IMobileGestures</i>)

Name	Description
PinchIn	Performs a two-pointer gesture, where each pointer moves toward the other, from the edges to the center of this object. For example to zoom out on an image. (Inherited from IMobileGestures)
PinchOut	Performs a two-pointer gesture, where each pointer moves opposite across the other, from the center out towards the edges of the this object. For example to zoom in on an image. (Inherited from IMobileGestures)
ScrollIntoView	Scroll until an object is visible. Works by dragging the scroll view in the indicated direction and then searching for the object, until either the object is found, or the repetition limit is reached (in which case an <code>ObjectNotFoundException</code> is thrown). (Inherited from MobileObject)
SendImeAction	Sends IME action to this object. (Inherited from MobileObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
Swipe	Performs a swipe between the two specified points. (Inherited from IMobileGestures)
SwipeDown	Performs a down-swipe. (Inherited from IMobileGestures)
SwipeLeft	Performs a left-swipe. (Inherited from IMobileGestures)
SwipeRight	Performs a right-swipe. (Inherited from IMobileGestures)
SwipeUp	Performs a up-swipe. (Inherited from IMobileGestures)
SwipeWithSegments	Performs a swipe between the specified points. (Inherited from IMobileGestures)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TwoPointerMultiTouch	Performs a multi-touch operation with two pointers. It executes two swipe operation at the same time. (Inherited from IMobileGestures)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IMobileKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Rumba Class Reference

Lists the available classes for testing Rumba controls.

Rumba Keys

Description

Contains the string constants for the Rumba key mnemonics which can be used in the SendKeys and SynchSendKeys methods of the RumbaObject class.

Includes the following string constants:

String Constant	Key Mnemonic
At	@@
Alt	@A
AltCursor	@e@1*
Attention	@A@Q
Backspace	@B
Backtab	@B
Clear	@C
Cmd	@A@Y
CursorDown	@V
CursorLeft	@L
CursorRight	@Z
CursorSelect	@e@1*
CursorUp	@U
Delete	@D
Dup	@S@x
End	@q
Enter	@E
EraseEof	@F
EraseEol	@e@d
EraseInput	@A@F
FieldExit	@A@E
FieldMark	@S@y
FieldMinus	@A@-
FieldPlus	@A@+
Help	@H
Hexadecimal	@A@X
Home	@0
Insert	@I
InsertToggle	@A@I
LocalHelp	@e@l
LocalPrint	@P
LocalHelpCursor	@e@m*
LeftTab	@B
NewLine	@e@n*
PageUp	@u

String Constant

PageDown
Print
PrintScreen
RecordBackspace
Reset
RightTab
Shift
SysRequest
Tab
Test
Usm
AplCutOnOf
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PF1
PF2
PF3
PF4
PF5
PF6
PF7
PF8
PF9
PF10
PF11
PF12
PF13
PF14

Key Mnemonic

@v
@A@t
@A@T
@A@<
@R
@T
@S
@A@H
@T
@A@C
@e@v
@e@2
@x
@y
@z
@+
@%
@&
@'
@(
@)
@*
@1
@2
@3
@4
@5
@6
@7
@8
@9
@a
@b
@c
@d
@e

String Constant	Key Mnemonic
PF15	@f
PF16	@g
PF17	@h
PF18	@i
PF19	@j
PF20	@k
PF21	@l
PF22	@m
PF23	@n
PF24	@o
Cents	@n@1
VerticalBar	@d@2
Corner	@d@3
Plus	@d@4
Minus	@d@5
Quote	@d@7
HorizontalBar	@d@8
OpenBracket	@d@9
CloseBracket	@d@10
Substitute	@d@0
Synchronous	@d@a

RumbaCharacterAttribute Data Type

Represents the character attributes of an on-screen character.

Can include the following character attributes:

Attribute	Description
backgroundColor	The background color of the character.
foregroundColor	The foreground color of the character.
column	The column of the character on the screen.
line	The line of the character on the screen.
autoEnter	Defines if the character is part of an auto-enter text.
autoSkip	Defines if the character is part of an auto-skip text.
blinking	Defines if the character is blinking.
bold	Defines if the character is bold.
columnSeparator	Defines if the character is a column separator.
detectable	Defines if the character is part of a detectable text.

Attribute	Description
hidden	Defines if the character is part of a hidden text.
protect	Defines if the character is part of a protected text.
reverseVideo	Defines if the character is displayed in reverse video.
underlined	Defines if the character is underlined.

For additional information, see *IRDEAttribute Properties* in the Rumba Developer Edition (RDE) documentation.

RumbaField Class

Description

Represents a field on the green screen.

Inheritance Hierarchy

- [RumbaObject](#)
 - RumbaField
 - [RumbaLabel](#)
 - [RumbaTextField](#)

Syntax

```
'Declaration
Public Class RumbaField _
Inherits RumbaObject
```

Properties

Name	Description
AutoEnter	A value that indicates true if the field is an AutoEnter field.
AutoSkip	A value that is true if the field is an AutoSkip field.
Detectable	A value is indicates true if the field is detectable.
Dup	A value that indicates true if the field is a dup field.
EndColumn	A value that indicates the ending column of the field.
EndLine	A value that indicates the ending line of the field.
EndPosition	The 1-based end position of the field, when the screen is seen as one sequence of characters.
Enptui	A value that indicates true if the field is an ENPTUI field.
FieldExitRequired	A value that indicates true if the field is required before exiting the screen.
Hidden	A value that indicates if the field is hidden.
Intense	A value that indicates true if the field is intense.
Length	The length of the field.

Name	Description
ModifiedDataTag	A value that indicates true if the field has the ModifiedDataTag property set.
Protected	A value that indicates true if the field is protected (i.e. a RumbaLabel).
StartColumn	A value that indicates the column in which the field begins.
StartLine	A value that indicates the line in which the field begins.
StartPosition	The 1-based start position of the field, when the screen is seen as one sequence of characters.
Text	The text of the control. (Inherited from TestObject)
Type	A value that represents the input allowed for this field.
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCharacterAttributes	Gets the character attributes for this field's characters.

Name	Description
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SendKeys	Sends keys. Supports RDE mnemonics. (Inherited from RumbaObject)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
SynchSendKeys	Synchronously sends keys. Supports RDE mnemonics. (Inherited from RumbaObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

RumbaLabel Class

Description

A field in the screen whose text can be only be retrieved but not set.

Inheritance Hierarchy

- [RumbaField](#)
 - RumbaLabel

Syntax

```
'Declaration
Public Class RumbaLabel _
Inherits RumbaField
```

Properties

Name	Description
AutoEnter	A value that indicates true if the field is an AutoEnter field. (Inherited from RumbaField)
AutoSkip	A value that is true if the field is an AutoSkip field. (Inherited from RumbaField)
Detectable	A value is indicates true if the field is detectable. (Inherited from RumbaField)

Name	Description
<i>Dup</i>	A value that indicates true if the field is a dup field. (Inherited from <i>RumbaField</i>)
<i>EndColumn</i>	A value that indicates the ending column of the field. (Inherited from <i>RumbaField</i>)
<i>EndLine</i>	A value that indicates the ending line of the field. (Inherited from <i>RumbaField</i>)
<i>EndPosition</i>	The 1-based end position of the field, when the screen is seen as one sequence of characters. (Inherited from <i>RumbaField</i>)
<i>Enptui</i>	A value that indicates true if the field is an ENPTUI field. (Inherited from <i>RumbaField</i>)
<i>FieldExitRequired</i>	A value that indicates true if the field is required before exiting the screen. (Inherited from <i>RumbaField</i>)
<i>Hidden</i>	A value that indicates if the field is hidden. (Inherited from <i>RumbaField</i>)
<i>Intense</i>	A value that indicates true if the field is intense. (Inherited from <i>RumbaField</i>)
<i>Length</i>	The length of the field. (Inherited from <i>RumbaField</i>)
<i>ModifiedDataTag</i>	A value that indicates true if the field has the ModifiedDataTag property set. (Inherited from <i>RumbaField</i>)
<i>Protected</i>	A value that indicates true if the field is protected (i.e. a RumbaLabel). (Inherited from <i>RumbaField</i>)
<i>StartColumn</i>	A value that indicates the column in which the field begins. (Inherited from <i>RumbaField</i>)
<i>StartLine</i>	A value that indicates the line in which the field begins. (Inherited from <i>RumbaField</i>)
<i>StartPosition</i>	The 1-based start position of the field, when the screen is seen as one sequence of characters. (Inherited from <i>RumbaField</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Type</i>	A value that represents the input allowed for this field. (Inherited from <i>RumbaField</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCharacterAttributes	Gets the character attributes for this field's characters. (Inherited from RumbaField)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SendKeys	Sends keys. Supports RDE mnemonics. (Inherited from RumbaObject)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
SynchSendKeys	Synchronously sends keys. Supports RDE mnemonics. (Inherited from RumbaObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

RumbaObject Class

Description
RumbaObject.

Inheritance Hierarchy

- [TestObject](#)
 - RumbaObject
 - [RumbaField](#)
 - [RumbaScreen](#)

Syntax

```
'Declaration  
Public Class RumbaObject _  
Inherits TestObject _  
Implements IClickable, IFocusable, IKeyable
```

Properties

Name	Description
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SendKeys	Sends keys. Supports RDE mnemonics.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
SynchSendKeys	Synchronously sends keys. Supports RDE mnemonics.
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

RumbaScreen Class

Description

Represents the whole green screen.

Inheritance Hierarchy

- [RumbaObject](#)
 - RumbaScreen

Syntax

```
'Declaration
Public Class RumbaScreen _
Inherits RumbaObject
```

Properties

Name	Description
BackgroundColor	the background color of the display control.
CharacterSetId	a value that identifies the character set translation table used by the Display control.
ColumnCount	a value that indicates the number of columns on the current screen.
CursorColumn	the column position of the cursor on the screen.
CursorLine	the line position of the cursor on the screen.
CursorPosition	The current cursor position on the screen. Values are starting with 1 for the first line or column.

Name	Description
HostConnected	a value that indicates if the host is connected.
HostDeviceName	the host device name.
HostIpAddress	the host IP address.
KeyboardLocked	a value that indicates whether keyboard input is inhibited.
LineCount	a value that indicates the number of lines on the current screen.
ScreenId	the screen id.
ScreenSize	the size of the screen in text lines and columns.
SessionName	the session name.
SupportsStructuredFields	a value that indicates if the host supports structured field decoding.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Clear	Clears the screen by sending the "clear" key. Equivalent to <code>synchSendKeys(RumbaKey.CLEAR)</code> .
Click	Clicks on the object. (Inherited from IClickable)
Connect	Connects to the hosts in the current configuration.
Disconnect	Disconnects the current connection.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetCharacterAttributes</i>	Gets the character attributes for length-number of characters starting at the given position.
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetContent</i>	Retrieves a string containing the screen without line breaks starting at the specified text position.
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetLines</i>	Retrieves a clipping of the screen and returns the selected lines in a list of strings.
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Reset	Resets the screen by sending the "reset" key. Equivalent to <code>synchSendKeys(RumbaKey.RESET)</code> .
SendKeys	Sends keys. Supports RDE mnemonics. (Inherited from RumbaObject)
SetCursorPosition	Moves the cursor to the given position of the screen.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
Submit	Submits the entered data to the server by sending the "submit" key. Equivalent to <code>synchSendKeys(RumbaKey.SUBMIT)</code> .
SynchSendKeys	Synchronously sends keys. Supports RDE mnemonics. (Inherited from RumbaObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForHostConnected	Waits until the session is connected.
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WriteText	Writes text on the screen.

RumbaTextField Class

Description

A field in the screen whose text can be both retrieved and set.

Inheritance Hierarchy

- [RumbaField](#)
 - RumbaTextField

Syntax

```
'Declaration  
Public Class RumbaTextField _  
Inherits RumbaField
```

Properties

Name	Description
AutoEnter	A value that indicates true if the field is an AutoEnter field. (Inherited from RumbaField)
AutoSkip	A value that is true if the field is an AutoSkip field. (Inherited from RumbaField)
Detectable	A value is indicates true if the field is detectable. (Inherited from RumbaField)
Dup	A value that indicates true if the field is a dup field. (Inherited from RumbaField)
EndColumn	A value that indicates the ending column of the field. (Inherited from RumbaField)
EndLine	A value that indicates the ending line of the field. (Inherited from RumbaField)
EndPosition	The 1-based end position of the field, when the screen is seen as one sequence of characters. (Inherited from RumbaField)
Enptui	A value that indicates true if the field is an ENPTUI field. (Inherited from RumbaField)
FieldExitRequired	A value that indicates true if the field is required before exiting the screen. (Inherited from RumbaField)
Hidden	A value that indicates if the field is hidden. (Inherited from RumbaField)
Intense	A value that indicates true if the field is intense. (Inherited from RumbaField)
Length	The length of the field. (Inherited from RumbaField)
ModifiedDataTag	A value that indicates true if the field has the ModifiedDataTag property set. (Inherited from RumbaField)
Protected	A value that indicates true if the field is protected (i.e. a RumbaLabel). (Inherited from RumbaField)
StartColumn	A value that indicates the column in which the field begins. (Inherited from RumbaField)
StartLine	A value that indicates the line in which the field begins. (Inherited from RumbaField)

Name	Description
StartPosition	The 1-based start position of the field, when the screen is seen as one sequence of characters. (Inherited from RumbaField)
Text	The text of the control. (Inherited from TestObject)
Type	A value that represents the input allowed for this field. (Inherited from RumbaField)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCharacterAttributes	Gets the character attributes for this field's characters. (Inherited from RumbaField)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SendKeys	Sends keys. Supports RDE mnemonics. (Inherited from RumbaObject)
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 SetText 	Sets the text field's text.
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 SynchSendKeys 	Synchronously sends keys. Supports RDE mnemonics. (Inherited from RumbaObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An

Name	Description
<i>WaitForPropertyNotEquals</i>	exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForScreenshotStable</i>	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

SAP Class Reference

Lists the available classes for testing SAP controls.

ISapContextMenuable Interface

Description

Interface for any Component which supports ContextMenu selections.

Syntax

```
'Declaration
Public Class ISapContextMenuable
```

Methods

Name	Description
<i>SelectContextMenuItem</i>	Select an item from the controls context menu.
<i>SelectContextMenuItemByPosition</i>	This method allows you to select a context menu item using the position of the item.
<i>SelectContextMenuItemByText</i>	Select a menu item of a context menu using the text of the item and possible higher level menus.

SapBarChart Class

Description

Class for SAP Bar Chart.

Inheritance Hierarchy

- [*SapShell*](#)

- SapBarChart

Syntax

```
'Declaration
Public Class SapBarChart _
Inherits SapShell _
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityDescription	the accessibility description of the shell.
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
ChartCount	the number of charts.
CurrentContextMenu	The current context menu. This is only set when a context menu is available at the shell object.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DragDropSupported	whether the shell allows drag and drop operations. (Inherited from SapShell)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)

Name	Description
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)
WindowHandle	the window handle of the control that is connected to the GuiShell.

Methods

Name	Description
BarCount	Returns the number of bars in the given chart.
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetBarContent	Returns the content of the bar.
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetGridLineContent	Returns the content of the grid line.

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GridCount	Returns the number of grids within the chart.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
LinkCount	Returns the number of links within the given chart.
SelectContextMenuItem	Select an item from the controls context menu. (Inherited from ISapContextMenuable)
SelectContextMenuItemByPosition	This method allows you to select a context menu item using the position of the item. (Inherited from ISapContextMenuable)
SelectContextMenuItemByText	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from ISapContextMenuable)
SendData	Send data to the server.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)

Name	Description
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapBox Class

Description

Class for SAP SapBox.

Inheritance Hierarchy

- [SapComponent](#)
 - SapBox

Syntax

```
'Declaration
Public Class SapBox _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)

Name	Description
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>ShowContextMenu</i>	Shows the context menu of the control. (Inherited from <i>SapComponent</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapButton Class

Description

SapButton class is the class for push buttons.

Inheritance Hierarchy

- [SapComponent](#)
 - SapButton

Syntax

```
'Declaration  
Public Class SapButton _  
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CharHeight	the height of the element in character metric.
CharLeft	the left coordinate of the element in character metric.
CharTop	the top coordinate of the element in character metric.
CharWidth	the width of the element in character metric.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)

Name	Description
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>Select</i>	This emulates manually pressing a button. Pressing a button will always cause server communication to occur, rendering all references to elements below the window level invalid.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>ShowContextMenu</i>	Shows the context menu of the control. (Inherited from <i>SapComponent</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapCalendar Class

Description

SapCalendar class represents a SapCalendar.

Inheritance Hierarchy

- [SapShell](#)
 - SapCalendar

Syntax

```
'Declaration  
Public Class SapCalendar _  
Inherits SapShell _  
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CurrentContextMenu	The current context menu. This is only set when a context menu is available at the shell object.
Day	The day of the month that is currently selected.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DragDropSupported	whether the shell allows drag and drop operations. (Inherited from SapShell)
EndSelection	The end of the selection interval.
FirstVisibleDate	the earliest date visible in the calendar control.
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)

Name	Description
LastVisibleDate	the last date visible in the calendar control.
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Month	The month of the year that is currently selected.
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
SelectionInterval	The interval that is currently selected.
StartSelection	The start of the selection interval.
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Week	The week of the year that is currently selected.
Width	the current width of the component. (Inherited from SapComponent)
Year	The year that is currently selected.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ContextMenu	Opens a context menu.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
SelectContextMenuItem	Select an item from the controls context menu. (Inherited from ISapContextMenuable)
SelectContextMenuItemByPosition	This method allows you to select a context menu item using the position of the item. (Inherited from ISapContextMenuable)
SelectContextMenuItemByText	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from ISapContextMenuable)
SetFirstVisibleDate	Sets the earliest date visible in the calendar control.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetFocusDate	Sets the date to be focused.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionInterval	Selects the specified interval.
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> ,

Name	Description
WaitForObject	because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapChart Class

Description

Class for SAP Chart.

Inheritance Hierarchy

- [SapShell](#)
 - SapChart

Syntax

```
'Declaration
Public Class SapChart _
Inherits SapShell _
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityDescription	the accessibility description of the shell.
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)

Name	Description
<i>AccessibilityTooltip</i>	an additional tooltip text for accessibility support. (Inherited from <i>SapComponent</i>)
<i>Changeable</i>	whether an object is changeable. (Inherited from <i>SapComponent</i>)
<i>CurrentContextMenu</i>	The current context menu. This is only set when a context menu is available at the shell object.
<i>DefaultTooltip</i>	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from <i>SapComponent</i>)
<i>DragDropSupported</i>	whether the shell allows drag and drop operations. (Inherited from <i>SapShell</i>)
<i>Height</i>	the current height of the component. (Inherited from <i>SapComponent</i>)
<i>IconName</i>	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from <i>SapComponent</i>)
<i>IsSymbolFont</i>	whether the component's text is visualized in the SAP symbol font. (Inherited from <i>SapComponent</i>)
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)
<i>WindowHandle</i>	the window handle of the control that is connected to the GuiShell.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SelectContextMenuItem	Select an item from the controls context menu. (Inherited from ISapContextMenuable)
SelectContextMenuItemByPosition	This method allows you to select a context menu item using the position of the item. (Inherited from ISapContextMenuable)
SelectContextMenuItemByText	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from ISapContextMenuable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
ValueChange	Changes a value of the chart.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

SapCheckBox Class

Description

SapCheckBox class is the class for check box controls.

Inheritance Hierarchy

- [SapComponent](#)
 - SapCheckBox

Syntax

```
'Declaration
Public Class SapCheckBox _
Inherits SapComponent
```

Properties

Name	Description
<i>AccessibilityText</i>	an additional text for accessibility support. (Inherited from <i>SapComponent</i>)
<i>AccessibilityTextOnRequest</i>	an additional text for accessibility support. (Inherited from <i>SapComponent</i>)
<i>AccessibilityTooltip</i>	an additional tooltip text for accessibility support. (Inherited from <i>SapComponent</i>)
<i>Changeable</i>	whether an object is changeable. (Inherited from <i>SapComponent</i>)
<i>CharHeight</i>	the height of the element in character metric.
<i>CharLeft</i>	the left coordinate of the element in character metric.
<i>CharTop</i>	the top coordinate of the element in character metric.
<i>CharWidth</i>	the width of the element in character metric.
<i>Checked</i>	whether the checkbox is checked.
<i>ColorIndex</i>	the index of the list color of this element.
<i>ColorIntensified</i>	whether the Intensified flag is set in screen painter for this element.
<i>ColorInverse</i>	whether the inverse color style is set in screen painter for the element.
<i>DefaultTooltip</i>	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from <i>SapComponent</i>)
<i>Flushing</i>	whether the value change causes a round trip.
<i>Height</i>	the current height of the component. (Inherited from <i>SapComponent</i>)
<i>IconName</i>	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from <i>SapComponent</i>)
<i>IsLeftLabel</i>	whether the component has the 'assign left' flag.
<i>IsListElement</i>	whether the element is on an ABAP list, not a screen.
<i>IsRightLabel</i>	whether the component has the 'assign right' flag.
<i>IsSymbolFont</i>	whether the component's text is visualized in the SAP symbol font. (Inherited from <i>SapComponent</i>)
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>RowText</i>	the text of the while line containing the current component.

Name	Description
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
Uncheck	Un-checks the check box.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapColorSelector Class

Description

SapColorSelector class represents a SapColorSelector.

Inheritance Hierarchy

- [SapShell](#)
 - SapColorSelector

Syntax

```
'Declaration  
Public Class SapColorSelector _  
Inherits SapShell _  
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CurrentContextMenu	The current context menu. This is only set when a context menu is available at the shell object.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DragDropSupported	whether the shell allows drag and drop operations. (Inherited from SapShell)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)

Name	Description
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeSelection	Selects the color at the given index position.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>SelectContextMenuItem</i>	Select an item from the controls context menu. (Inherited from <i>ISapContextMenuable</i>)
<i>SelectContextMenuItemByPosition</i>	This method allows you to select a context menu item using the position of the item. (Inherited from <i>ISapContextMenuable</i>)
<i>SelectContextMenuItemByText</i>	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from <i>ISapContextMenuable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>ShowContextMenu</i>	Shows the context menu of the control. (Inherited from <i>SapComponent</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapComboBox Class

Description

Class for SAP ComboBox.

Inheritance Hierarchy

- [SapComponent](#)
 - SapComboBox

Syntax

```
'Declaration  
Public Class SapComboBox _  
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CharHeight	the height of the element in character metric.
CharLeft	the left coordinate of the element in character metric.
CharTop	the top coordinate of the element in character metric.
CharWidth	the width of the element in character metric.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Flushing	whether the value change causes a round trip.
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsLeftLabel	whether the component has the 'assign left' flag.
IsListBoxActive	whether the list box of the the combo box is currently open.
IsRightLabel	whether the component has the 'assign right' flag.

Name	Description
<i>IsSymbolFont</i>	whether the component's text is visualized in the SAP symbol font. (Inherited from <i>SapComponent</i>)
<i>Key</i>	the key of the currently selected item.
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>Required</i>	whether the component is a required value for the screen.
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCurrentEntryKey	the key of the current entry.
GetCurrentEntryValue	the value of the current entry.
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetEntryKeys	a list of all entry keys.
GetEntryValues	a list of all entry values.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
SelectItem	Selects the specified item.
SelectKey	Selects the list item with the specified key.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapComponent Class

Description

SapComponent class is the base class for SAP.

Inheritance Hierarchy

- [TestObject](#)
 - SapComponent
 - [SapBox](#)
 - [SapButton](#)
 - [SapCheckBox](#)
 - [SapComboBox](#)
 - [SapContainer](#)
 - [SapContainerShell](#)
 - [SapCustomControl](#)
 - [SapDockShell](#)
 - [SapLabel](#)
 - [SapMenu](#)
 - [SapMenubar](#)
 - [SapOkCodeField](#)
 - [SapRadioButton](#)
 - [SapScrollContainer](#)
 - [SapShell](#)
 - [SapSimpleContainer](#)
 - [SapSplitterContainer](#)
 - [SapStatusbar](#)
 - [SapTab](#)
 - [SapTable](#)
 - [SapTabStrip](#)
 - [SapTextField](#)
 - [SapTitlebar](#)
 - [SapToolbar](#)

- [SapUserArea](#)
- [SapWindow](#)

Syntax

```
'Declaration
Public Class SapComponent _
Inherits TestObject _
Implements IFocusable
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support.
AccessibilityTextOnRequest	an additional text for accessibility support.
AccessibilityTooltip	an additional tooltip text for accessibility support.
Changeable	whether an object is changeable.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type.
Height	the current height of the component.
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string.
IsSymbolFont	whether the component's text is visualized in the SAP symbol font.
Left	the left position of the element in screen coordinates.
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system.
Name	the name of the component.
ScreenLeft	the left position of the component in screen coordinates.
ScreenTop	the top position of the component in screen coordinates.
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text.
Top	the top coordinate of the element in screen coordinates.
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	<p>parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
Exists	<p>Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)</p>
Exists	<p>Checks if an object exists in the application under test. (Inherited from TestObject)</p>
Find	<p>Finds an object specified by an XPath locator. (Inherited from TestObject)</p>
FindAll	<p>Finds all objects specified by an XPath locator. (Inherited from TestObject)</p>
GenerateLocator	<p>Returns a locator for this object. (Inherited from TestObject)</p>
GetChildren	<p>Returns the child objects of this object. (Inherited from TestObject)</p>
GetDynamicMethodList	<p>Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)</p>
GetParent	<p>Looks up the parent of this object in the test application. (Inherited from TestObject)</p>
GetProperty	<p>Returns the value of the specified property. (Inherited from TestObject)</p>
GetPropertyList	<p>Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)</p>
GetRect	<p>Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)</p>
HighlightObject	<p>Highlights this object. (Inherited from TestObject)</p>
ImageClick	<p>Clicks on specified image asset. (Inherited from TestObject)</p>
ImageClickFile	<p>Clicks on the specified image. (Inherited from TestObject)</p>
ImageExists	<p>Returns whether the specified image asset exists. (Inherited from TestObject)</p>
ImageExistsFile	<p>Returns whether the specified image exists. (Inherited from TestObject)</p>
ImageRectangle	<p>Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)</p>
ImageRectangleFile	<p>Returns the object-relative rectangle of the specified image. (Inherited from TestObject)</p>

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout

Name	Description
WaitForPropertyNotEquals	using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapContainer Class

Description

Class for SAP Container.

Inheritance Hierarchy

- [SapComponent](#)
 - SapContainer

Syntax

```
'Declaration
Public Class SapContainer _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)

Name	Description
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)

Name	Description
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapContainerShell Class

Description

Class for SAP ContainerShell.

Inheritance Hierarchy

- [SapComponent](#)
 - SapContainerShell

Syntax

```
'Declaration
Public Class SapContainerShell _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)

Name	Description
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>ShowContextMenu</i>	Shows the context menu of the control. (Inherited from <i>SapComponent</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapContextMenu Class

Description

The class for SAP context menus.

For information on determining the items of a SAP context menu, see [Determining the item names and item texts of a SapContextMenu](#).

Inheritance Hierarchy

- [SapMenu](#)
 - SapContextMenu

Syntax

```
'Declaration  
Public Class SapContextMenu _  
Inherits SapMenu
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)

Name	Description
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>Select</i>	Select the menu. (Inherited from SapMenu)
<i>SetFocus</i>	Gives focus to the control. (Inherited from IFocusable)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from TestObject)
<i>ShowContextMenu</i>	Shows the context menu of the control. (Inherited from SapComponent)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from TestObject)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from TestObject)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from TestObject)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from TestObject)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapCustomControl Class

Description

Class for SAP CustomControl.

Inheritance Hierarchy

- [SapComponent](#)
 - SapCustomControl

Syntax

```
'Declaration  
Public Class SapCustomControl _  
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)

Name	Description
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapDockShell Class

Description

Class for SAP DockShell.

Inheritance Hierarchy

- [SapComponent](#)
 - SapDockShell

Syntax

```
'Declaration
Public Class SapDockShell _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapGridView Class

Description

Class for SAP GridView. Some of the methods that have Get as a prefix require the column as an input parameter of type string. This parameter is different for each individual GridView element. To obtain the value of this parameter, record an action on the desired item and use the parameter that you have recorded.

Inheritance Hierarchy

- [SapShell](#)
 - SapGridView

Syntax

```
'Declaration
Public Class SapGridView _
Inherits SapShell _
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)

Name	Description
<i>AccessibilityTextOnRequest</i>	an additional text for accessibility support. (Inherited from <i>SapComponent</i>)
<i>AccessibilityTooltip</i>	an additional tooltip text for accessibility support. (Inherited from <i>SapComponent</i>)
<i>Changeable</i>	whether an object is changeable. (Inherited from <i>SapComponent</i>)
<i>ColumnCount</i>	the number of columns.
<i>ColumnOrder</i>	a list of strings which contains all the column identifiers in the order in which they are displayed.
<i>CurrentCellColumn</i>	the current column.
<i>CurrentCellRow</i>	the current row.
<i>CurrentContextMenu</i>	The current context menu. This is only set when a context menu is available at the shell object.
<i>DefaultTooltip</i>	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from <i>SapComponent</i>)
<i>DragDropSupported</i>	whether the shell allows drag and drop operations. (Inherited from <i>SapShell</i>)
<i>FirstVisibleColumn</i>	the first visible column of the scrollable area of the grid view, fixed columns are ignored.
<i>FirstVisibleRow</i>	the index of the first visible row in the grid.
<i>FrozenColumnCount</i>	the number of columns that are excluded from horizontal scrolling.
<i>Height</i>	the current height of the component. (Inherited from <i>SapComponent</i>)
<i>IconName</i>	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from <i>SapComponent</i>)
<i>IsSymbolFont</i>	whether the component's text is visualized in the SAP symbol font. (Inherited from <i>SapComponent</i>)
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>RowCount</i>	the number of rows.
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)

Name	Description
SelectedCells	a list of strings that contains the selected cells, each of which has the format "<row>,<column id>".
SelectedColumns	a list of strings of currently selected columns.
SelectedRows	a comma separated list of row index numbers or index ranges, such as "1,4,6-8".
SelectionMode	the selection mode. Possible values are: "RowsAndColumns", "ListboxSingle", "ListboxMultiple", "Free".
Text	The text of the control. (Inherited from TestObject)
Title	the title of the grid control.
ToolbarButtonCount	the number of tool bar buttons (separators included).
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
VisibleRowCount	the number of visible rows of the grid view.
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ClearSelection	Removes all row, column and cell selections.
Click	Clicks on the specified cell.
ClickCurrentCell	Clicks the current cell.
ContextMenu	Requests the context menu.
CurrentCellMoved	This function notifies the server that another cell was made the current cell. The function must be called whenever the current cell changes.
DeleteRows	Delete the given rows.
DeselectColumn	Removes the specified column from the list of selected columns.
DoubleClick	Double clicks the specified cell.

Name	Description
DoubleClickCurrentCell	Double clicks on the current cell.
DuplicateRows	Duplicates the given range of rows
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCellColor	Returns an identifier for the color of the cell.
GetCellHeight	Gets the height of the cell in pixels.
GetCellIcon	Gets the icon string of the specified cell.
GetCellLeft	Gets the left position of the cell.
GetCellMaxLength	Gets the maximum length of the cell in number of bytes.
GetCellState	Gets the state of the specified cell. Possible values are: Normal, Error, Warning, Info.
GetCellTooltip	Gets the tooltip of the specified cell.
GetCellTop	Gets the top position of the specified cell.
GetCellType	Gets the cell type of the specified cell.
GetCellValue	Gets the value of the specified cell.
GetCellWidth	Gets the width of the specified cell.
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetColorInfo	Returns the description for the color of the cell.
GetColumnDataType	Gets the data type of the column.
GetColumnPosition	Gets the position of the specified column.
GetColumnSortType	Gets the sort type of the column. Possible values are: None, Ascending, Descending.
GetColumnTitles	Gets all column titles of the specified column.
GetColumnTooltip	Gets the tool tip text of the specified column.
GetColumnTotalType	Gets the total type of the column. Possible values are: None, Total, Subtotal.
GetDisplayedColumnName	Gets the currently displayed title of the specified column.

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetRowTotalLevel</i>	Gets the level of the row.
<i>GetSymbolInfo</i>	Returns the description for the symbol in the cell.
<i>GetToolBarButtonIcon</i>	Gets the name of the icon of the specified toolbar button.
<i>GetToolBarButtonId</i>	Gets the ID of the specified tool bar button.
<i>GetToolBarButtonText</i>	Gets the text of the specified tool bar button.
<i>GetToolBarButtonTooltip</i>	Gets the tooltip of the specified tool bar button
<i>GetToolBarButtonType</i>	Gets the type of the specified toolbar button. Possible values are: "Button", "ButtonAndMenu", "Menu", "Separator", "Group", "CheckBox".
<i>GetToolBarFocusButton</i>	Gets the position of the current focused tool bar button
<i>HasCellF4Help</i>	Returns true if the specified cell has a value help assigned.
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>InsertRows</i>	Inserts a range of rows
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)

Name	Description
<i>IsCellChangeable</i>	Returns true if the specified cell is changeable.
<i>IsCellCheckBoxChecked</i>	Returns true if the check box at the specified position is checked.
<i>IsCellHotspot</i>	Returns true if the cell is a link.
<i>IsCellSymbol</i>	Returns true if the text in the cell is displayed in the SAP symbol font.
<i>IsCellTotalExpander</i>	Returns true if the specified cell contains a total expander button.
<i>IsColumnFiltered</i>	Returns true if the specified column is filtered.
<i>IsColumnKey</i>	Returns true if the specified column is a key column.
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>IsToolBarButtonChecked</i>	Returns true if the specified tool bar button is checked.
<i>IsToolBarButtonEnabled</i>	Returns true if the specified button is enabled.
<i>IsTotalRowExpanded</i>	whether the specified total row is expanded.
<i>ModifyCell</i>	Modifies the value of the specified cell.
<i>ModifyCheckBox</i>	Modifies the value of the specified checkbox cell.
<i>MoveRows</i>	Moves the rows with an index greater than or equal to fromRow up to an index less than or equal to toRow to the position of destRow.
<i>PressButton</i>	Presses the button placed in the specified cell.
<i>PressButtonCurrentCell</i>	Clicks a button placed in the current cell.
<i>PressColumnHeader</i>	Clicks on the header of the specified column.
<i>PressEnter</i>	Presses the enter key.
<i>PressF1</i>	Presses the F1 key.
<i>PressF4</i>	Presses the F4 key.
<i>PressToolBarButton</i>	Clicks the specified button in the grid view's tool bar.
<i>PressToolBarContextButton</i>	Opens the context menu of the specified button of the grid view's tool bar.
<i>PressTotalRow</i>	Presses the row button of the specified total row.
<i>PressTotalRowCurrentCell</i>	Presses the expansion button on the current cell.
<i>SelectAll</i>	Selects the whole grid content (i.e. all rows and all columns).
<i>SelectColumn</i>	Adds the specified column to the list of the selected columns.
<i>SelectContextMenuItem</i>	Select an item from the controls context menu. (Inherited from <i>ISapContextMenuable</i>)
<i>SelectContextMenuItemByPosition</i>	This method allows you to select a context menu item using the position of the item. (Inherited from <i>ISapContextMenuable</i>)

Name	Description
SelectContextMenuItemByText	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from ISapContextMenuable)
SelectionChanged	Notifies the server that the selection has changed
SelectToolBarMenuItem	Selects an item from the context menu of the grid view's tool bar.
SetColumnOrder	Sets the order of the columns.
SetColumnWidth	Sets the width of the specified column.
SetCurrentCell	Sets the specified cell as the current cell.
SetCurrentCellColumn	Specifies the column of the current cell.
SetCurrentCellRow	Sets the row of the current cell.
SetFirstVisibleColumn	Sets the first visible column.
SetFirstVisibleRow	Sets the first visible row.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectedRows	Selects the specified rows.
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TriggerModified	Notifies the server of multiple changes in cells. Typically this method should be called after multiple calls to <code>ModifyCell</code> .
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapHorizontalScrollBar Class

Description

Class for horizontal scrollbars.

Inheritance Hierarchy

- [SapScrollbar](#)
 - SapHorizontalScrollBar

Syntax

```
'Declaration
Public Class SapHorizontalScrollBar _
Inherits SapScrollbar
```

Properties

Name	Description
Maximum	the maximum scrollbar value. (Inherited from SapScrollbar)
Minimum	the minimum scrollbar value. (Inherited from SapScrollbar)
PageSize	the scrollbar pagesize. (Inherited from SapScrollbar)
Position	the current scrollbar position. (Inherited from SapScrollbar)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>ScrollTo</i>	Scrolls to the given position. (Inherited from <i>SapScrollbar</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapHTMLViewer Class

Description

Class for SAP HTMLView.

Inheritance Hierarchy

- [SapShell](#)
 - SapHTMLViewer

Syntax

```
'Declaration
Public Class SapHTMLViewer _
Inherits SapShell _
Implements ISapContextMenuable
```

Properties

Name	Description
<i>AccessibilityText</i>	an additional text for accessibility support. (Inherited from <i>SapComponent</i>)
<i>AccessibilityTextOnRequest</i>	an additional text for accessibility support. (Inherited from <i>SapComponent</i>)
<i>AccessibilityTooltip</i>	an additional tooltip text for accessibility support. (Inherited from <i>SapComponent</i>)
<i>Changeable</i>	whether an object is changeable. (Inherited from <i>SapComponent</i>)
<i>CurrentContextMenu</i>	The current context menu. This is only set when a context menu is available at the shell object.
<i>DefaultTooltip</i>	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from <i>SapComponent</i>)
<i>DocumentComplete</i>	whether the document is complete.
<i>DragDropSupported</i>	whether the shell allows drag and drop operations. (Inherited from <i>SapShell</i>)
<i>Height</i>	the current height of the component. (Inherited from <i>SapComponent</i>)
<i>IconName</i>	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from <i>SapComponent</i>)
<i>IsSymbolFont</i>	whether the component's text is visualized in the SAP symbol font. (Inherited from <i>SapComponent</i>)
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)

Name	Description
Width	the current width of the component. (Inherited from SapComponent)
WindowHandle	the window handle of the control that is connected to the GuiShell.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ContextMenu	Emulates the context menu request. Applies only to context menus provided by the backend.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SapEvent	This function submits an HTML form to the backend.
SelectContextMenuItem	Select an item from the controls context menu. (Inherited from ISapContextMenuable)
SelectContextMenuItemByPosition	This method allows you to select a context menu item using the position of the item. (Inherited from ISapContextMenuable)
SelectContextMenuItemByText	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from ISapContextMenuable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapLabel Class

Description

SapLabel class is the class for all SAP labels.

Inheritance Hierarchy

- [SapComponent](#)
 - SapLabel

Syntax

```
'Declaration  
Public Class SapLabel _  
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CharLeft	the left coordinate of the label in character metric.
CharTop	the top coordinate of the label in character metric.
CharWidth	the width of the element in character metric.
ColorIndex	the color index of the label.
ColorInverse	whether the inverse color style is set for the label.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DisplayedText	the text of the label.
Height	the current height of the component. (Inherited from SapComponent)
Highlighted	whether the label has the 'highlighted' flag assigned.
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsHotspot	whether a mouse click on the label causes a round trip.
IsLeftLabel	whether the component has the 'assign left' flag assigned.

Name	Description
IsListElement	whether the element is on an ABAP list.
IsRightLabel	whether the component has the 'assign right' flag assigned.
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Maxlength	the maximum text length of the label in code units.
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
Numerical	whether the label may only contain numbers.
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)

Name	Description
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapMenu Class

Description

Class for SAP Menu.

Inheritance Hierarchy

- [SapComponent](#)
 - SapMenu
 - [SapContextMenu](#)

Syntax

```
'Declaration
Public Class SapMenu _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)

Name	Description
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Select	Select the menu.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapMenubar Class

Description

Class for SAP MenuBar.

Inheritance Hierarchy

- [SapComponent](#)
 - SapMenubar

Syntax

```
'Declaration
Public Class SapMenubar _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)

Name	Description
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapNetPlan Class

Description

Class for SAP Net Chart.

Inheritance Hierarchy

- [SapShell](#)
 - SapNetPlan

Syntax

```
'Declaration  
Public Class SapNetPlan _  
Inherits SapShell _  
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityDescription	the accessibility description of the shell.
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CurrentContextMenu	The current context menu. This is only set when a context menu is available at the shell object.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DragDropSupported	whether the shell allows drag and drop operations. (Inherited from SapShell)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
LinkCount	the number of links in the net.
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
NodeCount	the number of Nodes in the net.

Name	Description
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)
WindowHandle	the window handle of the control that is connected to the GuiShell.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetLinkContent	Returns the content of the link.
GetNodeContent	Returns the content of the node.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SelectContextMenuItem	Select an item from the controls context menu. (Inherited from ISapContextMenuable)
SelectContextMenuItemByPosition	This method allows you to select a context menu item using the position of the item. (Inherited from ISapContextMenuable)
SelectContextMenuItemByText	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from ISapContextMenuable)
SendData	This function emulates the output of each action triggered at the control side. The result of the action is sent to the server.
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 ShowContextMenu 	Shows the context menu of the control. (Inherited from SapComponent)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForPropertyNotEquals 	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapOfficeIntegration Class

Description

Class for SAP OfficeIntegration.

Inheritance Hierarchy

- [SapShell](#)
 - SapOfficeIntegration

Syntax

```
'Declaration
Public Class SapOfficeIntegration _
Inherits SapShell _
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CurrentContextMenu	The current context menu. This is only set when a context menu is available at the shell object.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DragDropSupported	whether the shell allows drag and drop operations. (Inherited from SapShell)
Height	the current height of the component. (Inherited from SapComponent)

Name	Description
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>SelectContextMenuItem</i>	Select an item from the controls context menu. (Inherited from ISapContextMenuable)
<i>SelectContextMenuItemByPosition</i>	This method allows you to select a context menu item using the position of the item. (Inherited from ISapContextMenuable)

Name	Description
SelectContextMenuItemByText	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from ISapContextMenuable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An

Name	Description
WaitForPropertyNotEquals	exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapOkCodeField Class

Description

SapOkCodeField class is the class for sap OkCodeField.

Inheritance Hierarchy

- [SapComponent](#)
 - SapOkCodeField

Syntax

```
'Declaration
Public Class SapOkCodeField _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)

Name	Description
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
Opened	whether the GuiOkCodeField is collapsed.
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Close	Collapses the GuiOkCodeField.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Open	Expands the GuiOkCodeField.

Name	Description
PressF1	Emulate pressing the F1 key while the focus is on the GuiOkCodeField.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetText	Sets the text of the control to the specified text.
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An

Name	Description
WaitForPropertyNotEquals	exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapPicture Class

Description

Class for SAP picture.

Inheritance Hierarchy

- [SapShell](#)
 - SapPicture

Syntax

```
'Declaration
Public Class SapPicture _
Inherits SapShell _
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CurrentContextMenu	The current context menu. This is only set when a context menu is available at the shell object.

Name	Description
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DisplayMode	the display mode of the picture, i.e. "Normal", "Stretch", "Fit", "NormalCenter", "FitCenter"
DragDropSupported	whether the shell allows drag and drop operations. (Inherited from SapShell)
Height	the current height of the component. (Inherited from SapComponent)
Icon	the SAPGUI icon code e.g. ("@01@")
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Url	the URL of the picture
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	clicks on a picture
ClickControlArea	clicks on the specified position
ClickPictureArea	clicks on the specified position in the picture area
ContextMenu	The function opens a context menu on the given position. The coordinates should be given in pixels with respect to the picture control as it is displayed on the screen.
DoubleClick	double clicks on the picture
DoubleClickControlArea	double clicks on the specified position
DoubleClickPictureArea	double clicks on the specified position in the picture area
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>SelectContextMenuItem</i>	Select an item from the controls context menu. (Inherited from <i>ISapContextMenuable</i>)
<i>SelectContextMenuItemByPosition</i>	This method allows you to select a context menu item using the position of the item. (Inherited from <i>ISapContextMenuable</i>)
<i>SelectContextMenuItemByText</i>	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from <i>ISapContextMenuable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>ShowContextMenu</i>	Shows the context menu of the control. (Inherited from <i>SapComponent</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapRadioButton Class

Description

SapRadioButton class is the class for radio button controls.

Inheritance Hierarchy

- [SapComponent](#)
 - SapRadioButton

Syntax

```
'Declaration  
Public Class SapRadioButton _  
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CharHeight	the height of the element in character metric
CharLeft	the left coordinate of the element in character metric
CharTop	the top coordinate of the element in character metric
CharWidth	the width of the element in character metric
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Flushing	whether the value change causes a round trip
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsLeftLabel	whether the component has the 'assign left' flag
IsRightLabel	whether the component has the 'assign right' flag
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)

Name	Description
Selected	whether the radio button is selected
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Select	Selects the radio button.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using

Name	Description
WaitForDisappearance	the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapScrollbar Class

Description

Class for SAP Scrollbar.

Inheritance Hierarchy

- [TestObject](#)
 - SapScrollbar

- [SapHorizontalScrollBar](#)
- [SapVerticalScrollBar](#)

Syntax

```
'Declaration
Public Class SapScrollbar _
Inherits TestObject
```

Properties

Name	Description
Maximum	the maximum scrollbar value.
Minimum	the minimum scrollbar value.
PageSize	the scrollbar pagesize.
Position	the current scrollbar position.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>ScrollTo</i>	Scrolls to the given position.
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapScrollContainer Class

Description

Class for SAP UserArea.

Inheritance Hierarchy

- [SapComponent](#)
 - SapScrollContainer

Syntax

```
'Declaration  
Public Class SapScrollContainer _  
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)

Name	Description
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapShell Class

Description

Class for SAP SapShell.

Inheritance Hierarchy

- [SapComponent](#)
 - SapShell
 - [SapBarChart](#)
 - [SapCalendar](#)
 - [SapChart](#)
 - [SapColorSelector](#)
 - [SapGridView](#)
 - [SapHTMLViewer](#)
 - [SapNetPlan](#)

- [SapOfficeIntegration](#)
- [SapPicture](#)
- [SapTextEdit](#)
- [SapToolbarControl](#)
- [SapTree](#)

Syntax

```
'Declaration
Public Class SapShell _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DragDropSupported	whether the shell allows drag and drop operations.
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)

Name	Description
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>ShowContextMenu</i>	Shows the context menu of the control. (Inherited from <i>SapComponent</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapSimpleContainer Class

Description

Class for SAP SimpleContainer.

Inheritance Hierarchy

- [SapComponent](#)
 - SapSimpleContainer

Syntax

```
'Declaration  
Public Class SapSimpleContainer _  
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapSplitterContainer Class

Description

Class for SAP splitter container.

Inheritance Hierarchy

- [SapComponent](#)
 - SapSplitterContainer

Syntax

```
'Declaration
Public Class SapSplitterContainer _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)

Name	Description
<i>Changeable</i>	whether an object is changeable. (Inherited from <i>SapComponent</i>)
<i>CurrentContextMenu</i>	The current context menu. This is only set when a context menu is available at the shell object.
<i>DefaultTooltip</i>	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from <i>SapComponent</i>)
<i>Height</i>	the current height of the component. (Inherited from <i>SapComponent</i>)
<i>IconName</i>	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from <i>SapComponent</i>)
<i>IsSymbolFont</i>	whether the component's text is visualized in the SAP symbol font. (Inherited from <i>SapComponent</i>)
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An

Name	Description
WaitForPropertyNotEquals	exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapStatusbar Class

Description

Class for SAP statusbar.

Inheritance Hierarchy

- [SapComponent](#)
 - SapStatusbar

Syntax

```
'Declaration
Public Class SapStatusbar _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)

Name	Description
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
MessageId	the name of the message class used in the ABAP message call
MessageNumber	the message number used in the ABAP message call
MessageType	the type of the displayed message in the statusbar, i.e. S (Success), W (Warning), E (Error), A (Abort), I (Information)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
DoubleClick	double-clicks the statusbar
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>ShowContextMenu</i>	Shows the context menu of the control. (Inherited from <i>SapComponent</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForObject</i>	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from <i>TestObject</i>)
<i>WaitForProperty</i>	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapTab Class

Description

Class for SAP Tab.

Inheritance Hierarchy

- [SapComponent](#)
 - SapTab

Syntax

```
'Declaration
Public Class SapTab _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)

Name	Description
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Select	Selects the tab
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)

Name	Description
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapTable Class

Description

SapTable class is the class for SAP tables.

Inheritance Hierarchy

- [SapComponent](#)
 - SapTable

Syntax

```
'Declaration
Public Class SapTable _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CharHeight	the height of the element in character metric.
CharLeft	the left coordinate of the element in character metric.
CharTop	the top coordinate of the element in character metric.
CharWidth	the width of the element in character metric.
ColumnCount	the number of columns in the table.
ColumnSelectionMode	the column selection mode
CurrentColumn	the index of the current column.
CurrentRow	the index of the current row.

Name	Description
<i>DefaultTooltip</i>	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from <i>SapComponent</i>)
<i>Height</i>	the current height of the component. (Inherited from <i>SapComponent</i>)
<i>IconName</i>	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from <i>SapComponent</i>)
<i>IsSymbolFont</i>	whether the component's text is visualized in the SAP symbol font. (Inherited from <i>SapComponent</i>)
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>RowCount</i>	the number of rows in the table including invisible rows.
<i>RowSelectionMode</i>	the row selection mode.
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>TableFieldName</i>	the table field name.
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>VisibleRowCount</i>	the number of visible rows in the table.
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ConfigureLayout	Opens the configuration layout dialog, in which the layout of the table can be changed.
DeselectAllColumns	This function can be used for table controls with a button that allows to de-select all columns in one step.
DeselectColumn	Deselects the given column.
DeselectRow	Deselects the given row.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCell	Returns a visible table cell.
GetCellAbsolute	Returns a table cell. If the row is not visible, it will be scrolled into view.
GetCellsForRow	Returns the cells for the given row.
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetColumnName	This method returns the title of the given column.
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVisibleCellsForColumn	Returns the visible cells for the given column.

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsColumnSelected	Returns true if the given column is selected.
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
IsRowSelected	Returns true if the given row is selected.
ReorderTable	This method re-orders the columns.
SelectAllColumns	This function can be used for table controls with a button that allows to select all columns in one step.
SelectColumn	Selects the given column.
SelectRow	Selects the given row.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapTabStrip Class

Description

Class for SAP tab pane.

Inheritance Hierarchy

- [SapComponent](#)
 - SapTabStrip

Syntax

```
'Declaration  
Public Class SapTabStrip _  
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
SelectedTab	the current selected tab.
Tabs	all tabs of the tab pane.
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)

Name	Description
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapTextEdit Class

Description

Class for SAP TextEdit.

Inheritance Hierarchy

- [SapShell](#)
 - SapTextEdit

Syntax

```
'Declaration  
Public Class SapTextEdit _  
Inherits SapShell
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
CurrentColumn	the column index in which the caret is currently positioned.
CurrentContextMenu	The current context menu. This is only set when a context menu is available at the shell object.
CurrentLine	the line index in which the caret is currently positioned.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DragDropSupported	whether the shell allows drag and drop operations. (Inherited from SapShell)
FirstVisibleLine	the first visible line which is visualized at the top border of the text edit control.
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
LastVisibleLine	the last visible line which is visualized at the bottom border of the text edit control.
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
LineCount	the number of all lines in the text edit control.
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)

Name	Description
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
SelectedText	the currently selected text.
SelectionEndColumn	the column index in which the selection ends.
SelectionEndLine	the line index in which the selection ends.
SelectionIndexEnd	the overall end index of the text selection.
SelectionIndexStart	the overall start index of the text selection.
SelectionStartColumn	the column start index of the text selection.
SelectionStartLine	the line index in which the selection starts.
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ContextMenu	Requests the context menu.
DoubleClick	Double clicks the text edit control.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetLineText	Returns the text of the specified line.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsBreakpointLine	whether the specified line contains a breakpoint.
IsCommentLine	Returns true if the specified line is a comment line.
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
IsHighlightedLine	Returns true if the specified line is highlighted.

Name	Description
<i>IsProtectedLine</i>	Returns true if the specified line is protected.
<i>PressF1</i>	Presses the F1 key.
<i>PressF4</i>	Presses the F4 key.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>SetSelectionIndexes</i>	Selects the text within the specified range.
<i>SetText</i>	Sets the specified text in the text edit control.
<i>ShowContextMenu</i>	Shows the context menu of the control. (Inherited from <i>SapComponent</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForObject</i>	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from <i>TestObject</i>)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapTextField Class

Description

SapTextField is the class for single-line and multi-line fields whose text can be modified by the user.

Inheritance Hierarchy

- [SapComponent](#)
 - SapTextField

Syntax

```
'Declaration
Public Class SapTextField _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)

Name	Description
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DisplayedText	the text as it is displayed on the screen, including preceding or trailing blanks.
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsPassword	whether the text field is a password text field.
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetPasswordText	Sends an encrypted password to the object.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetText	Sets the text of the control to the specified text.
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapTitlebar Class

Description

Class for SAP Titlebar.

Inheritance Hierarchy

- [SapComponent](#)
 - SapTitlebar

Syntax

```
'Declaration
Public Class SapTitlebar _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)

Name	Description
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is

Name	Description
	passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter,

Name	Description
WaitForScreenshotStable	<p>the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

SapToolbar Class

Description

Class for SAP Toolbar.

Inheritance Hierarchy

- [SapComponent](#)
 - SapToolbar

Syntax

```
'Declaration
Public Class SapToolbar _
Inherits SapComponent
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)

Name	Description
<i>IsSymbolFont</i>	whether the component's text is visualized in the SAP symbol font. (Inherited from <i>SapComponent</i>)
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)

Name	Description
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
	OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapToolbarControl Class

Description

Class for SAP ToolbarControl.

Inheritance Hierarchy

- [SapShell](#)
 - SapToolbarControl

Syntax

```
'Declaration
Public Class SapToolbarControl _
Inherits SapShell _
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
ButtonCount	the number of toolbar buttons including separators.
Changeable	whether an object is changeable. (Inherited from SapComponent)
CurrentContextMenu	The current context menu. This is only set when a context menu is available at the shell object.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DragDropSupported	whether the shell allows drag and drop operations. (Inherited from SapShell)
FocusedButton	the zero-based index of the button that currently has the focus.
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)

Name	Description
<i>IsSymbolFont</i>	whether the component's text is visualized in the SAP symbol font. (Inherited from <i>SapComponent</i>)
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)
<i>WindowHandle</i>	the window handle of the control that is connected to the GuiShell.

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetButtonIcon	the name of the icon of the specified toolbar button.
GetButtonType	the type of the specified toolbar button. Possible values are: "Button", "ButtonAndMenu", "Menu", "Separator", "Group", "CheckBox"
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsButtonChecked	if the button is currently checked (pressed).
IsButtonEnabled	if the button can be pressed.

Name	Description
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
PressContextButton	Press the context button with the given id.
SelectButton	emulates pressing the button with the given id.
SelectContextMenuItem	Select an item from the controls context menu. (Inherited from ISapContextMenuable)
SelectContextMenuItemByPosition	This method allows you to select a context menu item using the position of the item. (Inherited from ISapContextMenuable)
SelectContextMenuItemByText	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from ISapContextMenuable)
SelectMenuItem	emulates selecting the menu item with the given id.
SelectMenuItemByText	emulates selecting the menu item by menu item text.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapTree Class

Description

SapTree class is the class for all SAP trees. Some of the methods that have Get as a prefix require the nodeKey and the ItemName as input parameters of type string. These parameters are different for each individual SapTree element. To obtain the value of these parameters, record an action on the desired item and use the parameters that you have recorded.

Inheritance Hierarchy

- [SapShell](#)
 - SapTree

Syntax

```
'Declaration
Public Class SapTree _
Inherits SapShell _
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityDescription	the accessibility description of the shell.
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)
ColumnHeaders	a collection of the titles of the columns.
ColumnNames	a collection of the column names.
ColumnOrder	the column order.
ColumnTitles	a collection of the titles of the columns.
CurrentContextMenu	The current context menu. This is only set when a context menu is available at the shell object.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
DragDropSupported	whether the shell allows drag and drop operations. (Inherited from SapShell)
FocusedNodeKey	the key of the focused node.
Height	the current height of the component. (Inherited from SapComponent)
HierarchyHeaderWidth	the width of the hierarchy header in pixels.
HierarchyTitle	the hierarchy title.
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
NodeKeys	a collection containing either the node key of the root node, or the node keys of all nodes in the tree, depending on the application.

Name	Description
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
SelectedItemColumn	the key of the currently selected item column.
SelectedItemNode	the key of the currently selected item node.
SelectedNode	the key of the currently selected node.
SelectedNodes	a collection containing the node keys of all the selected nodes in the tree.
SelectionMode	the selection mode.
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
TopNode	the key of the top node.
TreeType	the tree type. (simple tree, list tree or column tree)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ChangeCheckbox	This method emulates changing a checkbox state.
CollapseNode	This function closes the node with the key nodeKey.
DefaultContextMenu	This method requests a context menu for the whole Tree Control.
DoubleSelectedItem	This function emulates double clicking on a text item.
DoubleSelectNode	This function emulates double clicking a node.
EnsureVisibleHorizontalItem	This function scrolls the Tree horizontally until the Item is visible.

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
ExpandNode	This function expands the node with the key <i>nodeKey</i> .
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
FindNodeKeyByPath	Return the node key for the given path (e.g. 2¥1¥2).
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetAbapImage	Gets the abap image.
GetCheckBoxState	Retrieves the CheckBox state (true = Checked, false = Unchecked).
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetColumnIndexFromName	Gets the column index for the given name.
GetColumnKeys	The keys of all the items in the given column.
GetColumnNameFromName	Gets the column title for the given name.
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHierarchyLevel	Gets the hierarchy level for the given key.
GetItemHeight	Returns the current item height.
GetItemLeft	The left position of the component.
GetItemStyle	The style of the given item.
GetItemText	For multi column trees, this function returns the text of the item specified by the key and name parameters.
GetItemTextColor	Returns the item text color.
GetItemTooltip	Returns the item tooltip text.
GetItemTop	The top position of the component.
GetItemType	Returns the type of the given node.
GetItemWidth	Returns the current item width.
GetListTreeNodeItemCount	Returns the node item count.
GetNextNodeKey	Returns the key of the next node.

Name	Description
GetNodeAbapImage	Returns the abap image for the given node.
GetNodeChildrenCount	This function returns the number of children of the node given.
GetNodeChildrenCountByPath	This function returns the number of children of the node given by the path parameter (e.g. 2¥1¥2).
GetNodeHeight	The current height of the node.
GetNodeIndex	The index of the given node.
GetNodeItemHeaders	Returns the headers for the given node.
GetNodeLeft	The left position of the component.
GetNodePathByKey	Given a node key, the path is retrieved (e.g. 2¥1¥2).
GetNodeStyle	Returns the style of the node.
GetNodeText	This function returns the text of the node specified by the given key.
GetNodeTextByPath	The text of a node defined by the given path is returned.
GetNodeTextColor	Returns the node text color.
GetNodeTooltip	Returns the node tooltip text.
GetNodeTop	The top position of the component.
GetNodeWidth	Returns the current node width.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetParentNodeKey	Key of the parent node of the node specified by the given key.
GetPreviousNodeKey	Key of the previous node of the node specified by the given key.
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetStyleDescription	Returns the description for the given style.
GetSubNodesKeys	Collection of the keys of all subnodes of the node specified by the given key.
HeaderContextMenu	This method requests a context menu for a header.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>IsFolder</i>	Whether the given node is a folder.
<i>IsFolderExpandable</i>	Whether the folder is expandable.
<i>IsFolderExpanded</i>	Whether the folder is expanded.
<i>IsHighLighted</i>	Whether the node is highlighted.
<i>ItemContextMenu</i>	This method requests a context menu for an item.
<i>NodeContextMenu</i>	This method requests a context menu for a node.
<i>PressKey</i>	This method emulates pressing a key.
<i>SelectButton</i>	This method emulates pressing a button.
<i>SelectColumn</i>	This function adds a column to the column selection. A node or item selection is removed.
<i>SelectContextMenuItem</i>	Select an item from the controls context menu. (Inherited from <i>ISapContextMenuable</i>)
<i>SelectContextMenuItemByPosition</i>	This method allows you to select a context menu item using the position of the item. (Inherited from <i>ISapContextMenuable</i>)
<i>SelectContextMenuItemByText</i>	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from <i>ISapContextMenuable</i>)
<i>SelectHeader</i>	This method emulates clicking a header.
<i>SelectItem</i>	This function emulates the selection of an item. This selection removes all other selections.
<i>SelectLink</i>	This function emulates triggering a link.
<i>SelectNode</i>	The node with the key nodeKey is added to the Node Selection.
<i>SetCheckBoxState</i>	Changes the state of the checkbox in the given node.
<i>SetColumnOrder</i>	Sets the order of the tree's column.
<i>SetColumnWidth</i>	This function sets the width of a column in pixels.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)

Name	Description
SetHierarchyHeaderWidth	Sets the width of the Hierarchy Header in pixels.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectedNode	Select the given node.
SetTopNode	Sets the given key as top node.
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
UnselectAll	All selections are removed.
UnselectColumn	This function removes a column from the column selection.
UnselectNode	The node with the key nodeKey is removed from the Node Selection.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapUserArea Class

Description

Class for SAP UserArea.

Inheritance Hierarchy

- [SapComponent](#)
 - SapUserArea

Syntax

```
'Declaration
Public Class SapUserArea _
Inherits SapComponent _
Implements ISapContextMenuable
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTooltip	an additional tooltip text for accessibility support. (Inherited from SapComponent)
Changeable	whether an object is changeable. (Inherited from SapComponent)

Name	Description
CurrentContextMenu	The current context menu. This is only set when a context menu is available at the shell object.
DefaultTooltip	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from SapComponent)
Height	the current height of the component. (Inherited from SapComponent)
IconName	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from SapComponent)
IsSymbolFont	whether the component's text is visualized in the SAP symbol font. (Inherited from SapComponent)
Left	the left position of the element in screen coordinates. (Inherited from SapComponent)
Modified	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from SapComponent)
Name	the name of the component. (Inherited from SapComponent)
ScreenLeft	the left position of the component in screen coordinates. (Inherited from SapComponent)
ScreenTop	the top position of the component in screen coordinates. (Inherited from SapComponent)
Text	The text of the control. (Inherited from TestObject)
Tooltip	the tooltip text. (Inherited from SapComponent)
Top	the top coordinate of the element in screen coordinates. (Inherited from SapComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	the current width of the component. (Inherited from SapComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>SelectContextMenuItem</i>	Select an item from the controls context menu. (Inherited from <i>ISapContextMenuable</i>)
<i>SelectContextMenuItemByPosition</i>	This method allows you to select a context menu item using the position of the item. (Inherited from <i>ISapContextMenuable</i>)
<i>SelectContextMenuItemByText</i>	Select a menu item of a context menu using the text of the item and possible higher level menus. (Inherited from <i>ISapContextMenuable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>ShowContextMenu</i>	Shows the context menu of the control. (Inherited from <i>SapComponent</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForObject</i>	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapVerticalScrollBar Class

Description

Class for vertical scrollbars.

Inheritance Hierarchy

- [SapScrollbar](#)
 - SapVerticalScrollBar

Syntax

```
'Declaration
Public Class SapVerticalScrollBar _
Inherits SapScrollbar
```

Properties

Name	Description
Maximum	the maximum scrollbar value. (Inherited from SapScrollbar)
Minimum	the minimum scrollbar value. (Inherited from SapScrollbar)
PageSize	the scrollbar pagesize. (Inherited from SapScrollbar)
Position	the current scrollbar position. (Inherited from SapScrollbar)
Text	The text of the control. (Inherited from TestObject)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
ScrollTo	Scrolls to the given position. (Inherited from SapScrollbar)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By

Name	Description
	default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SapWindow Class

Description

SapWindow class is the class for a top-level window.

Inheritance Hierarchy

- [SapComponent](#)
 - SapWindow

Syntax

```
'Declaration
Public Class SapWindow _
Inherits SapComponent _
Implements IMoveable
```

Properties

Name	Description
AccessibilityText	an additional text for accessibility support. (Inherited from SapComponent)
AccessibilityTextOnRequest	an additional text for accessibility support. (Inherited from SapComponent)

Name	Description
<i>AccessibilityTooltip</i>	an additional tooltip text for accessibility support. (Inherited from <i>SapComponent</i>)
<i>ButtonbarVisible</i>	whether the application toolbar, the lower toolbar within SAP GUI, is visible.
<i>Changeable</i>	whether an object is changeable. (Inherited from <i>SapComponent</i>)
<i>DefaultTooltip</i>	the tooltip text which is generated from the short text defined in the data dictionary for the given screen element type. (Inherited from <i>SapComponent</i>)
<i>GuiFocus</i>	the element within SAP GUI that currently has the focus.
<i>Height</i>	the current height of the component. (Inherited from <i>SapComponent</i>)
<i>IconName</i>	whether the object has been assigned an icon, then this property is the name of the icon, otherwise it is an empty string. (Inherited from <i>SapComponent</i>)
<i>IsSymbolFont</i>	whether the component's text is visualized in the SAP symbol font. (Inherited from <i>SapComponent</i>)
<i>Left</i>	the left position of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Modified</i>	whether the state has been changed by the user and this change has not yet been sent to the SAP system. (Inherited from <i>SapComponent</i>)
<i>Name</i>	the name of the component. (Inherited from <i>SapComponent</i>)
<i>ScreenLeft</i>	the left position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>ScreenTop</i>	the top position of the component in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>StatusbarVisible</i>	whether the statusbar is visible.
<i>SystemFocus</i>	the element within SAP GUI that has the focus from the server application's perspective.
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>TitlebarVisible</i>	whether the titlebar is visible.
<i>ToolbarVisible</i>	whether the toolbar is visible.
<i>Tooltip</i>	the tooltip text. (Inherited from <i>SapComponent</i>)
<i>Top</i>	the top coordinate of the element in screen coordinates. (Inherited from <i>SapComponent</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>Width</i>	the current width of the component. (Inherited from <i>SapComponent</i>)

Name	Description
WindowHandle	the window handle of the control that is connected to the GuiShell.
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)
WorkingPaneHeight	the current height of the working pane.
WorkingPaneWidth	the current width of the working pane.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
<i>GetFocus</i>	Returns the object with the input focus. (Inherited from <i>IMoveable</i>)
<i>GetNextCloseWindow</i>	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from <i>IMoveable</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsActive</i>	Returns whether the window is set to active. (Inherited from <i>IMoveable</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>IsVkeyAllowed</i>	The method returns True if the given virtual key can be executed on the current screen.
<i>JumpBackward</i>	Execute the Ctrl+Shift+Tab key on the window to jump backward one block.
<i>JumpForward</i>	Execute the Ctrl+Tab key on the window to jump forward one block.
<i>Maximize</i>	Maximizes the window. (Inherited from <i>IMoveable</i>)
<i>Minimize</i>	Reduces the window to an icon. (Inherited from <i>IMoveable</i>)

Name	Description
Move	Moves the window. (Inherited from IMoveable)
ResizeWorkingPane	The ResizeWorkingPane function will resize the window so that the available working area has the given width and height in character metric.
ResizeWorkingPaneEx	The ResizeWorkingPaneEx function will resize the window so that the available working area has the given width and height in pixels.
Restore	Restores the window to its previous size. (Inherited from IMoveable)
SendVKey	Send a virtual key to the system.
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
ShowContextMenu	Shows the context menu of the control. (Inherited from SapComponent)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TabBackward	Execute the Shift+Tab key on the window to jump backward one element.
TabForward	Execute the Tab key on the window to jump forward one element.
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Silverlight Class Reference

Lists the available classes for testing Silverlight controls.

SLApplication Class

Description

The container for Silverlight controls.

Inheritance Hierarchy

- [SLBase](#)
 - SLApplication

Syntax

```
'Declaration  
Public Class SLApplication _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLAutoCompleteBox Class

Description

Represents a control that provides a text box for user input and a drop-down that contains possible matches based on the input in the text box.

Inheritance Hierarchy

- [SLBase](#)
 - SLAutoCompleteBox

Syntax

```
'Declaration
Public Class SLAutoCompleteBox _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)

Name	Description
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
HorizontallyScrollable	a value that indicates whether the combo box can scroll horizontally.
HorizontalScrollPercent	the current horizontal scroll position or negative one (-1) if there is no valid scroll position.
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsOpen	a value that indicates whether the auto complete box is open
Items	a list containing items of the current matches. If nothing was entered in the control's text box an empty list is returned.
ItemTexts	a list containing the texts of the current matches. If nothing was entered in the control's text box an empty list is returned.
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
VerticallyScrollable	a value that indicates whether the combo box can scroll vertically.
VerticalScrollPercent	the current vertical scroll position or negative one (-1) if there is no valid scroll position.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the drop-down list that contains the matches if it is currently open.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollHorizontal</i>	Scrolls the currently visible region of the content area, horizontally, the specified ScrollAmount.
<i>ScrollVertical</i>	Scrolls the currently visible region of the content area, vertically, the specified ScrollAmount.
<i>Select</i>	Types the given text into the auto box complete box and selects a match.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetHorizontalScrollPercent</i>	Sets the horizontal scroll position as a percentage of the total content area within the combo box.
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>SetVerticalScrollPercent</i>	Sets the vertical scroll position as a percentage of the total content area within the combo box.
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLBase Class

Description

Provides a Silverlight framework-level set of properties, events, and methods for Silverlight elements.

Inheritance Hierarchy

- [TestObject](#)
 - SLBase
 - [SLApplication](#)
 - [SLAutoCompleteBox](#)
 - [SLButton](#)
 - [SLCalendar](#)
 - [SLCalendarButton](#)
 - [SLCalendarDayButton](#)
 - [SLCheckBox](#)
 - [SLComboBox](#)
 - [SLComboBoxItem](#)
 - [SLDataGrid](#)
 - [SLDataGridCell](#)
 - [SLDataGridDetails](#)
 - [SLDataGridRow](#)
 - [SLDataPager](#)
 - [SLDatePicker](#)
 - [SLDescriptionViewer](#)
 - [SLFrame](#)
 - [SLGridSplitter](#)
 - [SLGroup](#)
 - [SLHeader](#)
 - [SLHeaderItem](#)
 - [SLHorizontalScrollBar](#)
 - [SLHyperlinkButton](#)
 - [SLImage](#)
 - [SLListBox](#)
 - [SLListItem](#)
 - [SLMediaElement](#)
 - [SLMenu](#)
 - [SLMenuBar](#)
 - [SLMenuItem](#)
 - [SLMultiScaleImage](#)
 - [SLPane](#)
 - [SLPasswordBox](#)
 - [SLPopup](#)
 - [SLProgressBar](#)
 - [SLRadioButton](#)
 - [SLRepeatButton](#)
 - [SLRichTextBox](#)

- [SLSeparator](#)
- [SLSlider](#)
- [SLSpinner](#)
- [SLSplitButton](#)
- [SLStatusBar](#)
- [SLTabControl](#)
- [SLTabItem](#)
- [SLTable](#)
- [SLTextBlock](#)
- [SLTextBox](#)
- [SLThumb](#)
- [SLTitleBar](#)
- [SLToggleButton](#)
- [SLToolBar](#)
- [SLToolTip](#)
- [SLTreeView](#)
- [SLTreeViewItem](#)
- [SLValidationSummary](#)
- [SLVerticalScrollBar](#)
- [SLWindow](#)

Syntax

```
'Declaration
Public Class SLBase _
Inherits TestObject _
Implements IClickable, IFocusable, IKeyable
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element.
ClassName	the simple class name of the element.
ControlType	a string containing the control type for the element.
HelpText	a string containing the help text (e.g. tooltip text) for the element.
IsEnabled	a value that indicates whether the element is enabled.
Name	a string containing the UI Automation name for the element.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default

Name	Description
	<p>bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type.
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLButton Class

Description

Identifies a button control.

Inheritance Hierarchy

- [SLBase](#)
 - SLButton

Syntax

```
'Declaration  
Public Class SLButton _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Clicks the button.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLCalendar Class

Description

Identifies a calendar control.

Inheritance Hierarchy

- [SLBase](#)
 - SLCalendar

Syntax

```
'Declaration  
Public Class SLCalendar _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
CanSelectMultiple	whether multiple dates can be selected
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
DisplayMode	a value that indicates whether the calendar displays a month, year, or decade.
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsSelectionRequired	whether at least one date has to be selected
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
SelectedDate	the currently selected date in the currently displayed view.
SelectedDates	a list of selected dates in the currently displayed view.
SupportedDisplayModes	a list of supported display modes.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SelectDate	Selects the given date.
SelectDates	Selects the given dates.
SelectRange	Selects a range of dates.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
SwitchToDecadeView	Switches the calendar display mode to 'decade', if supported.
SwitchToMonthView	Switches the calendar display mode to 'month', if supported.
SwitchToYearView	Switches the calendar display mode to 'year', if supported.
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLCalendarButton Class

Description

Identifies a calendar button.

Inheritance Hierarchy

- [SLBase](#)
 - SLCalendarButton

Syntax

```
'Declaration
Public Class SLCalendarButton _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsSelected	a value that indicates whether the calendar button is selected
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Select</i>	Clicks the button.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLCalendarDayButton Class

Description

Identifies a calendar day button.

Inheritance Hierarchy

- [SLBase](#)
 - SLCalendarDayButton

Syntax

```
'Declaration  
Public Class SLCalendarDayButton _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsSelected	a value that indicates whether the calendar day button is selected
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Clicks the button.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLCheckBox Class

Description

Identifies a check box control.

Inheritance Hierarchy

- [SLBase](#)
 - SLCheckBox

Syntax

```
'Declaration  
Public Class SLCheckBox _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
ToggleState	the toggle state of the check box
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetIndeterminate	Set the check box to the indeterminate state.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
Toggle	Cycles through the toggle states of the check box in this order: On, Off and, if supported, Indeterminate.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks the check box.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using

Name	Description
	the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLComboBox Class

Description

Identifies a combo box control.

Combo boxes in Silverlight require to be opened once before information about their items can be retrieved.

Inheritance Hierarchy

- [SLBase](#)
 - SLComboBox

Syntax

```
'Declaration  
Public Class SLComboBox _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
HorizontallyScrollable	a value that indicates whether the combo box can scroll horizontally.
HorizontalScrollPercent	the current horizontal scroll position or negative one (-1) if there is no valid scroll position.
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsOpen	a value that indicates whether the combo box is open
Items	a complete list of items.
ItemTexts	a complete list of item texts.
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
SelectedIndex	the index of the selected item or negative one (-1) if the selection is empty.
SelectedItem	the selected item or Nothing if the selection is empty.
SelectedItemText	the text of the selected item or an empty string if the selection is empty.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
VerticallyScrollable	a value that indicates whether the combo box can scroll vertically.
VerticalScrollPercent	the current vertical scroll position or negative one (-1) if there is no valid scroll position.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the combo box.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Open	Opens the combo box.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollHorizontal	Scrolls the currently visible region of the content area, horizontally, the specified ScrollAmount.
ScrollVertical	Scrolls the currently visible region of the content area, vertically, the specified ScrollAmount.
Select	Selects an item in the combo box.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetHorizontalScrollPercent	Sets the horizontal scroll position as a percentage of the total content area within the combo box.

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 SetVerticalScrollPercent 	Sets the vertical scroll position as a percentage of the total content area within the combo box.
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLComboBoxItem Class

Description

Identifies a combo box item control.

Inheritance Hierarchy

- [SLBase](#)
 - SLComboBoxItem

Syntax

```
'Declaration
Public Class SLComboBoxItem _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsSelected	a value that indicates whether the combo box item is selected
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be

Name	Description
	specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects the combo box item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLDataGrid Class

Description

Identifies a data grid control.

If the data grid has a vertical scroll bar, rows which are not in the visible area of the data grid do not expose any information (e.g. text, caption). In order to interact with such a row or retrieve information about the row it needs to be scrolled into view first.

Inheritance Hierarchy

- [SLBase](#)
 - SLDataGrid

Syntax

```
'Declaration
Public Class SLDataGrid _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
CanSelectMultiple	whether the data grid allows more than one row to be selected concurrently.
ClassName	the simple class name of the element. (Inherited from SLBase)
ColumnCount	the number of columns in the data grid
ColumnHeaders	a list containing the header item for every column
ColumnHeaderTexts	a list containing the text of the header item for every column
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
HorizontallyScrollable	a value that indicates whether the data grid can scroll horizontally.
HorizontalScrollPercent	the current horizontal scroll position or negative one (-1) if there is no valid scroll position.

Name	Description
<i>IsEnabled</i>	a value that indicates whether the element is enabled. (Inherited from <i>SLBase</i>)
<i>Name</i>	a string containing the UI Automation name for the element. (Inherited from <i>SLBase</i>)
<i>RowCount</i>	the number of rows in the data grid
<i>Rows</i>	a list of rows in the data grid
<i>RowTexts</i>	a list of row texts in the data grid. The text for every row is returned as a string that contains the text of every cell separated with a blank character.
<i>SelectedCell</i>	the currently selected cell or Nothing if no cell is selected
<i>SelectedCellText</i>	the text of the currently selected cell or an empty string if no cell is selected
<i>SelectedRow</i>	the currently selected row or Nothing if no row is selected
<i>SelectedRows</i>	a list of currently selected rows or an empty list if no row is selected
<i>SelectedRowText</i>	the text of the currently selected row or an empty string if no row is selected
<i>SelectedRowTexts</i>	the text of the currently selected rows or an empty list if no row is selected
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)
<i>VerticallyScrollable</i>	a value that indicates whether the data grid can scroll vertically.
<i>VerticalScrollPercent</i>	the current vertical scroll position or negative one (-1) if there is no valid scroll position.

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollHorizontal</i>	Scrolls the currently visible region of the content area, horizontally, the specified ScrollAmount.
<i>ScrollIntoView</i>	Scrolls the specified row or cell into view.
<i>ScrollVertical</i>	Scrolls the currently visible region of the content area, vertically, the specified ScrollAmount.
<i>Select</i>	Selects the specified row.
<i>SelectCell</i>	Selects the specified cell.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetHorizontalScrollPercent</i>	Sets the horizontal scroll position as a percentage of the total content area within the data grid.
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>SetVerticalScrollPercent</i>	Sets the vertical scroll position as a percentage of the total content area within the data grid.
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLDataGridCell Class

Description

Identifies a cell within a row of a data grid control.

Inheritance Hierarchy

- [SLBase](#)
 - SLDataGridCell

Syntax

```
'Declaration  
Public Class SLDataGridCell _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsSelected	a value that indicates whether the cell is selected
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scroll the cell into view.
Select	Selects the cell
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLDataGridDetails Class

Description

Identifies a control that displays additional details within a data grid row.

Inheritance Hierarchy

- [SLBase](#)
 - SLDataGridDetails

Syntax

```
'Declaration  
Public Class SLDataGridDetails _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLDataGridRow Class

Description

Identifies a row in data grid.

Inheritance Hierarchy

- [SLBase](#)
 - SLDataGridRow

Syntax

```
'Declaration
Public Class SLDataGridRow _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
Cells	the cells of the data grid row.
CellTexts	the text of the cells of the data grid row.
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsSelected	a value that indicates whether the row is selected.
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
RowHeader	the header for this row
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the row into view.
Select	Selects the row
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLDataPager Class

Description

Identifies a data pager for paging through a collection of data.

Inheritance Hierarchy

- [SLBase](#)
 - SLDataPager

Syntax

```
'Declaration
Public Class SLDataPager _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
CurrentPage	The current page.
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
PageCount	The count of the pages.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
MoveToFirstPage	Moves the data pager to the first page.
MoveToLastPage	Moves the data pager to the last page.
MoveToNextPage	Moves the data pager to the next page.
MoveToPage	Moves the data pager to the given page.
MoveToPreviousPage	Moves the data pager to the previous page.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLDatePicker Class

Description

Represents a control that allows the user to select a date.

Inheritance Hierarchy

- [SLBase](#)
 - SLDatePicker

Syntax

```
'Declaration
Public Class SLDatePicker _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsCalendarOpen	a value whether the drop-down Calendar is open.
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
SelectedDate	the selected date or Nothing if no date is selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
CloseCalendar	Closes the drop-down Calendar.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenCalendar	Opens the drop-down Calendar.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change

Name	Description
	the timeout by setting the value of the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLDescriptionViewer Class

Description

Identifies a description viewer control.

Inheritance Hierarchy

- [SLBase](#)
 - SLDescriptionViewer

Syntax

```
'Declaration
Public Class SLDescriptionViewer _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)

Name	Description
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLFrame Class

Description

Identifies a frame control.

Inheritance Hierarchy

- [SLBase](#)
 - SLFrame

Syntax

```
'Declaration
Public Class SLFrame _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)

Name	Description
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLGridSplitter Class

Description

Identifies a grid splitter control.

Inheritance Hierarchy

- [SLBase](#)
 - SLGridSplitter

Syntax

```
'Declaration
Public Class SLGridSplitter _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MoveHorizontal	Moves the splitter horizontally.
MoveVertical	Moves the splitter vertically.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLGroup Class

Description

Identifies a group control, which acts as a container for other controls.

Inheritance Hierarchy

- [SLBase](#)
 - SLGroup

Syntax

```
'Declaration
Public Class SLGroup _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLHeader Class

Description

Identifies a control that provides a visual container for the labels for rows or columns.

Inheritance Hierarchy

- [SLBase](#)
 - SLHeader

Syntax

```
'Declaration  
Public Class SLHeader _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HeaderItems	a list containing the header items of this header control
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is

Name	Description
	passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLHeaderItem Class

Description

Identifies a control that provides a visual label for a row or column.

Inheritance Hierarchy

- [SLBase](#)
 - SLHeaderItem

Syntax

```
'Declaration
Public Class SLHeaderItem _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject . (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects the header item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLHorizontalScrollBar Class

Description

Identifies a horizontal scroll bar.

Inheritance Hierarchy

- [SLBase](#)
 - SLHorizontalScrollBar

Syntax

```
'Declaration
Public Class SLHorizontalScrollBar _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)

Name	Description
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Maximum	the maximum scroll position
Minimum	the minimum scroll position
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
ScrollPosition	the current scroll position
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Scrolls the scroll bar by the specified amount
ScrollToMaximum	Scrolls the scroll bar to the right end.
ScrollToMinimum	Scrolls the scroll bar to the left end.
ScrollToPosition	Scrolls the scroll bar to the specific position
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> ,

Name	Description
WaitForObject	because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLHyperlinkButton Class

Description

Identifies a hyperlink button control.

Inheritance Hierarchy

- [SLBase](#)
 - SLHyperlinkButton

Syntax

```
'Declaration
Public Class SLHyperlinkButton _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)

Name	Description
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Clicks the hyperlink button.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout

Name	Description
<i>WaitForPropertyNotEquals</i>	using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
<i>WaitForScreenshotStable</i>	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLImage Class

Description

Identifies an image control.

Inheritance Hierarchy

- [SLBase](#)
 - SLImage

Syntax

```
'Declaration
Public Class SLImage _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)

Name	Description
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

SLListBox Class

Description

Identifies a list box control.

Inheritance Hierarchy

- [SLBase](#)
 - SLListBox

Syntax

```
'Declaration
Public Class SLListBox _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
CanSelectMultiple	whether the list box allows more than one child element to be selected concurrently.
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
HorizontallyScrollable	a value that indicates whether the list box can scroll horizontally.
HorizontalScrollPercent	the current horizontal scroll position or negative one (-1) if there is no valid scroll position.
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Items	the list of list box items.
ItemTexts	the list of all item texts.

Name	Description
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
SelectedIndex	the index of the first item in the current selection or returns negative one (-1) if the selection is empty.
SelectedIndices	the indices of the currently selected items. An empty list is returned if no items are currently selected.
SelectedItem	the first item in the current selection or returns Nothing if the selection is empty
SelectedItems	the currently selected items. An empty list is returned, if no items are currently selected.
SelectedItemText	the text of the first selected item.
SelectedItemTexts	the texts of the currently selected items. An empty list is returned if no items are currently selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
VerticallyScrollable	a value that indicates whether the list box can scroll vertically.
VerticalScrollPercent	the current vertical scroll position or negative one (-1) if there is no valid scroll position.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DeselectAll	Deselects all list items.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollHorizontal	Scrolls the currently visible region of the content area, horizontally, the specified ScrollAmount.
ScrollVertical	Scrolls the currently visible region of the content area, vertically, the specified ScrollAmount.
Select	Selects the given list box item.
SelectAll	Selects all items in the list.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetHorizontalScrollPercent	Sets the horizontal scroll position as a percentage of the total content area within the list box.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetVerticalScrollPercent	Sets the vertical scroll position as a percentage of the total content area within the list box.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using

Name	Description
	the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLListItem Class

Description

Identifies a list item control.

Inheritance Hierarchy

- [SLBase](#)
 - SLListItem

Syntax

```
'Declaration  
Public Class SListItem _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsSelected	a value that indicates whether the list item is selected
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Deselect	Deselects the item.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)

Name	Description
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scroll the list item into view.
<i>Select</i>	Selects the list item.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLMediaElement Class

Description

Represents a control that contains audio and/or video.

Inheritance Hierarchy

- [SLBase](#)
 - SLMediaElement

Syntax

```
'Declaration
Public Class SLMediaElement _
Inherits SLBase
```

Properties

Name	Description
<i>AutomationId</i>	a string containing the UI Automation identifier (ID) for the element. (Inherited from <i>SLBase</i>)
<i>ClassName</i>	the simple class name of the element. (Inherited from <i>SLBase</i>)
<i>ControlType</i>	a string containing the control type for the element. (Inherited from <i>SLBase</i>)
<i>HelpText</i>	a string containing the help text (e.g. tooltip text) for the element. (Inherited from <i>SLBase</i>)
<i>IsEnabled</i>	a value that indicates whether the element is enabled. (Inherited from <i>SLBase</i>)
<i>Name</i>	a string containing the UI Automation name for the element. (Inherited from <i>SLBase</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject . (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLMenu Class

Description

Identifies a menu control.

Inheritance Hierarchy

- [SLBase](#)
 - SLMenu

Syntax

```
'Declaration
Public Class SLMenu _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)

Name	Description
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForPropertyNotEquals 	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLMenuBar Class

Description

Identifies a menu bar control.

Inheritance Hierarchy

- [SLBase](#)
 - SLMenuBar

Syntax

```
'Declaration
Public Class SLMenuBar _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLMenuItem Class

Description

Identifies a menu item control.

Inheritance Hierarchy

- [SLBase](#)
 - SLMenuItem

Syntax

```
'Declaration  
Public Class SLMenuItem _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForProperty	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLMultiScaleImage Class

Description

Enables users to open a multi-resolution image, which can be zoomed in on and panned across.

Inheritance Hierarchy

- [SLBase](#)
 - SLMultiScaleImage

Syntax

```
'Declaration  
Public Class SLMultiScaleImage _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLPane Class

Description

Identifies a pane that can optionally have scroll bars.

Inheritance Hierarchy

- [SLBase](#)
 - SLPane

Syntax

```
'Declaration
Public Class SLPane _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)

Name	Description
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
HorizontallyScrollable	a value that indicates whether the pane can scroll horizontally.
HorizontalScrollPercent	the current horizontal scroll position or negative one (-1) if there is no valid scroll position.
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
VerticallyScrollable	a value that indicates whether the pane can scroll vertically.
VerticalScrollPercent	the current vertical scroll position or negative one (-1) if there is no valid scroll position.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollHorizontal	Scrolls the currently visible region of the content area, horizontally, the specified ScrollAmount.
ScrollVertical	Scrolls the currently visible region of the content area, vertically, the specified ScrollAmount.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetHorizontalScrollPercent	Sets the horizontal scroll position as a percentage of the total content area within the pane.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetVerticalScrollPercent	Sets the vertical scroll position as a percentage of the total content area within the pane.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLPasswordBox Class

Description

Identifies a password box control.

Inheritance Hierarchy

- [SLBase](#)
 - SLPasswordBox

Syntax

```
'Declaration
Public Class SLPasswordBox _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsReadOnly	whether the password box is read-only
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all text from the password field.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetPosition	Sets the insertion point in the password field.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelection	Selects a range of text in the password box.
SetText	Replaces the text in the password field with the given text.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change

Name	Description
	the timeout by setting the value of the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLPopup Class

Description

Identifies a popup control. Context menus are also treated as popup controls.

Inheritance Hierarchy

- [SLBase](#)
 - SLPopup

Syntax

```
'Declaration
Public Class SLPopup _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)

Name	Description
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLProgressBar Class

Description

Identifies a progress bar control.

Inheritance Hierarchy

- [SLBase](#)
 - SLProgressBar

Syntax

```
'Declaration
Public Class SLProgressBar _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)

Name	Description
<i>Maximum</i>	the maximum progress value
<i>Minimum</i>	the minimum progress value
<i>Name</i>	a string containing the UI Automation name for the element. (Inherited from <i>SLBase</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)

Name	Description
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForPropertyNotEquals 	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLRadioButton Class

Description

Represents a button that allows a user to select a single option from a group of options.

Inheritance Hierarchy

- [SLBase](#)
 - SLRadioButton

Syntax

```
'Declaration
Public Class SLRadioButton _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsSelected	a value that indicates whether the radio button is selected.
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Deselects any selected items and then selects the current element.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLRepeatButton Class

Description

Identifies a repeat button control.

Inheritance Hierarchy

- [SLBase](#)
 - SLRepeatButton

Syntax

```
'Declaration
Public Class SLRepeatButton _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Clicks the button.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLRichTextBox Class

Description

Identifies a rich text box control.

Inheritance Hierarchy

- [SLBase](#)
 - SLRichTextBox

Syntax

```
'Declaration  
Public Class SLRichTextBox _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
SelectedText	the selected text
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all text from the text box.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetPosition	Sets the insertion point in the text box.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelection	Selects a range of text in the text box.
SetText	Replaces the content in the text box with the given text.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLSeparator Class

Description

Identifies a separator control.

Inheritance Hierarchy

- [SLBase](#)
 - SLSeparator

Syntax

```
'Declaration  
Public Class SLSeparator _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLSlider Class

Description

Identifies a progress bar control.

Inheritance Hierarchy

- [SLBase](#)
 - SLSlider

Syntax

```
'Declaration
Public Class SLSlider _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)

Name	Description
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
LargeChange	the value that is added to or subtracted from the Value property when a large change is made, such as with the PAGE DOWN key.
Maximum	the maximum progress value
Minimum	the minimum progress value
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
SmallChange	the value that is added to or subtracted from the Value property when a small change is made, such as with an arrow key.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DecreaseLarge	Decreases the value of the Slider by a large amount.
DecreaseSmall	Decreases the value of the Slider by a small amount.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>IncreaseLarge</i>	Increases the value of the Slider by a large amount.
<i>IncreaseSmall</i>	Increases the value of the Slider by a small amount.
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>SetValue</i>	Sets the value of the slider
<i>SetValueToMaximum</i>	Sets the Slider to the maximum position.
<i>SetValueToMinimum</i>	Sets the Slider to the minimum position.
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLSpinner Class

Description

Identifies a spinner control.

A spinner control type consists of a set of buttons that enable a user to select from a set of items or set a numerical value from within a range.

Inheritance Hierarchy

- [SLBase](#)
 - SLSpinner

Syntax

```
'Declaration  
Public Class SLSpinner _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
NumericValue	the current numeric value of the spinner
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLSplitButton Class

Description

Identifies a split button control.

Inheritance Hierarchy

- [SLBase](#)
 - SLSplitButton

Syntax

```
'Declaration
Public Class SLSplitButton _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)

Name	Description
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLStatusBar Class

Description

Identifies a status bar control.

Inheritance Hierarchy

- [SLBase](#)
 - SLStatusBar

Syntax

```
'Declaration
Public Class SLStatusBar _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)

Name	Description
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

SLTabControl Class

Description

Identifies a tab control.

Inheritance Hierarchy

- [SLBase](#)
 - SLTabControl

Syntax

```
'Declaration
Public Class SLTabControl _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Items	a complete list of tab items.
ItemTexts	a complete list of tab item texts.
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
SelectedIndex	the index of the selected item or negative one (-1) if the selection is empty.
SelectedItem	the selected item or Nothing if the selection is empty.

Name	Description
SelectedItemText	the text of the selected item or an empty string if the selection is empty.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRawChildren</i>	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from <i>SLBase</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Select</i>	Selects an item in the tab control.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForPropertyNotEquals 	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLTabItem Class

Description

Identifies a tab item control.

Inheritance Hierarchy

- [SLBase](#)
 - SLTabItem

Syntax

```
'Declaration
Public Class SLTabItem _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsSelected	a value that indicates whether the tab item is selected
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects the tab item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLTable Class

Description

Identifies a table.

Inheritance Hierarchy

- [SLBase](#)
 - SLTable

Syntax

```
'Declaration
Public Class SLTable _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLTextBlock Class

Description

Identifies a control for displaying small amounts of flow content.

Inheritance Hierarchy

- [SLBase](#)
 - SLTextBlock

Syntax

```
'Declaration  
Public Class SLTextBlock _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLTextBox Class

Description

Identifies a text box control.

Inheritance Hierarchy

- [SLBase](#)
 - SLTextBox

Syntax

```
'Declaration
Public Class SLTextBox _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsReadOnly	whether the text box is read-only
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all text from the text field.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetPosition	Sets the insertion point in the text field.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelection	Selects a range of text in the text box.
SetText	Replaces the text in the text field with the given text.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change

Name	Description
	the timeout by setting the value of the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Examples

```
textBox.SetText("some text")
textBox.SetSelection(0, 4) ' selects 4 characters starting from the first one
textBox.TypeKeys("<Ctrl+C>") ' copies the selected text to the clipboard
Dim selectedText = Clipboard.GetText() ' gets the text from the clipboard
```

SLThumb Class

Description

Identifies a thumb control.

Inheritance Hierarchy

- [SLBase](#)
 - SLThumb

Syntax

```
'Declaration
Public Class SLThumb _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLTitleBar Class

Description

Identifies the caption bar on a window.

Inheritance Hierarchy

- [SLBase](#)
 - SLTitleBar

Syntax

```
'Declaration
Public Class SLTitleBar _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)

Name	Description
<i>IsEnabled</i>	a value that indicates whether the element is enabled. (Inherited from <i>SLBase</i>)
<i>Name</i>	a string containing the UI Automation name for the element. (Inherited from <i>SLBase</i>)
<i>Text</i>	The text of the control. (Inherited from <i>TestObject</i>)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from <i>TestObject</i>)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from <i>TestObject</i>)
<i>Click</i>	Clicks on the object. (Inherited from <i>IClickable</i>)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from <i>IClickable</i>)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from <i>TestObject</i>)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForPropertyNotEquals 	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLToggleButton Class

Description

Identifies a toggle button control.

Inheritance Hierarchy

- [SLBase](#)
 - SLToggleButton

Syntax

```
'Declaration
Public Class SLToggleButton _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
ToggleState	the toggle state of the button
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
Toggle	Cycles through the toggle states of the button in this order: On, Off and, if supported, Indeterminate.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLToolBar Class

Description

Identifies a toolbar control.

Inheritance Hierarchy

- [SLBase](#)
 - SLToolBar

Syntax

'Declaration

```
Public Class SLToolBar _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLToolTip Class

Description

Identifies a tooltip control.

Inheritance Hierarchy

- [SLBase](#)
 - SLToolTip

Syntax

```
'Declaration  
Public Class SLToolTip _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLTreeView Class

Description

Represents a control that displays hierarchical data in a tree structure that has items that can expand and collapse.

Inheritance Hierarchy

- [SLBase](#)
 - SLTreeView

Syntax

```
'Declaration
Public Class SLTreeView _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
HorizontallyScrollable	a value that indicates whether the tree view can scroll horizontally.
HorizontalScrollPercent	the current horizontal scroll position or negative one (-1) if there is no valid scroll position.
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
RootItems	a complete list of all root tree items.
RootItemTexts	a complete list of the texts of all root tree items.
SelectedItem	the selected item or returns Nothing if the selection is empty.
SelectedItemPath	the selected item path or an empty item path if nothing is selected
SelectedItemText	the text of the selected item or an empty string if the selection is empty.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
VerticallyScrollable	a value that indicates whether the tree view can scroll vertically.
VerticalScrollPercent	the current vertical scroll position or negative one (-1) if there is no valid scroll position.
VisibleItemPaths	the visible items in the tree as item paths.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Collapse	Collapses the specified item.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Expand	Expands the specified item.
ExpandAll	Expands all items in the tree.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollHorizontal</i>	Scrolls the currently visible region of the content area, horizontally, the specified ScrollAmount.
<i>ScrollVertical</i>	Scrolls the currently visible region of the content area, vertically, the specified ScrollAmount.
<i>Select</i>	Selects the specified item.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetHorizontalScrollPercent</i>	Sets the horizontal scroll position as a percentage of the total content area within the tree view.
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>SetVerticalScrollPercent</i>	Sets the vertical scroll position as a percentage of the total content area within the tree view.
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLTreeViewItem Class

Description

Represents a selectable item in a tree control.

Inheritance Hierarchy

- [SLBase](#)
 - SLTreeViewItem

Syntax

```
'Declaration
Public Class SLTreeViewItem _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
IsExpanded	a value that indicates whether the nested items in a tree item are expanded or collapsed.
IsLeaf	a value that indicates whether the tree view item is a leaf item.
IsSelected	a value that indicates whether a tree item is selected.
Items	a complete list of items.
ItemTexts	a complete list of item texts.
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Collapse	Collapses the item.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Expand	Expands the item.
ExpandSubTree	Expands the sub tree under this tree item.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the tree item into view
Select	Selects the item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForPropertyNotEquals 	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLValidationSummary Class

Description

Identifies a validation summary control.

Inheritance Hierarchy

- [SLBase](#)
 - SLValidationSummary

Syntax

```
'Declaration
Public Class SLValidationSummary _
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLVerticalScrollBar Class

Description

Identifies a vertical scroll bar.

Inheritance Hierarchy

- [SLBase](#)
 - SLVerticalScrollBar

Syntax

```
'Declaration  
Public Class SLVerticalScrollBar _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Maximum	the maximum scroll position
Minimum	the minimum scroll position
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
ScrollPosition	the current scroll position
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel

Name	Description
	must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Scroll	Scrolls the scroll bar by the specified amount
ScrollToMaximum	Scrolls the scroll bar to the bottom.
ScrollToMinimum	Scrolls the scroll bar to the top.
ScrollToPosition	Scrolls the scroll bar to the specific position
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SLWindow Class

Description

Identifies a window control.

Inheritance Hierarchy

- [SLBase](#)
 - SLWindow

Syntax

```
'Declaration  
Public Class SLWindow _  
Inherits SLBase
```

Properties

Name	Description
AutomationId	a string containing the UI Automation identifier (ID) for the element. (Inherited from SLBase)
ClassName	the simple class name of the element. (Inherited from SLBase)
ControlType	a string containing the control type for the element. (Inherited from SLBase)
HelpText	a string containing the help text (e.g. tooltip text) for the element. (Inherited from SLBase)
IsEnabled	a value that indicates whether the element is enabled. (Inherited from SLBase)
Name	a string containing the UI Automation name for the element. (Inherited from SLBase)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRawChildren	Gets the raw child element of this element. This method can be used to access elements that would otherwise not be recognized as children. Optionally one of the MSUIA control types can be specified to only return children of a certain type. (Inherited from SLBase)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Java AWT and Swing Class Reference

Lists the available classes for testing Java AWT and Swing controls.

AbstractButton Class

Description

Defines common behaviors for buttons and menu items.

Inheritance Hierarchy

- [JComponent](#)
 - AbstractButton
 - [JButton](#)
 - [JMenuItem](#)
 - [JToggleButton](#)

Syntax

```
'Declaration  
Public Class AbstractButton _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)

Name	Description
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button.
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForPropertyNotEquals 	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Applet Class

Description

Represents the Java Applet in the browser.

Inheritance Hierarchy

- [AWTContainer](#)
 - Applet

Syntax

```
'Declaration
Public Class Applet _
Inherits AWTContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)

Name	Description
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AppletContainer Class

Description

Root element of an Applet in the browser.

Inheritance Hierarchy

- [AWTContainer](#)
 - AppletContainer

Syntax

```
'Declaration
Public Class AppletContainer _
Inherits AWTContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)

Name	Description
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> ,

Name	Description
WaitForObject	because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTButton Class

Description

Represents a push button.

Inheritance Hierarchy

- [AWTComponent](#)
 - AWTButton

Syntax

```
'Declaration
Public Class AWTButton _
Inherits AWTComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)

Name	Description
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from AWTComponent)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTCanvas Class

Description

Represents an empty rectangular area of the screen where the application can draw or trap events from the user.

Inheritance Hierarchy

- [AWTComponent](#)
 - AWTCanvas

Syntax

```
'Declaration
Public Class AWTCanvas _
Inherits AWTComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)

Name	Description
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTCheckbox Class

Description

Represents a check box.

Inheritance Hierarchy

- [AWTComponent](#)
 - AWTCheckbox

Syntax

```
'Declaration
Public Class AWTCheckbox _
Inherits AWTComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Checked	whether the check box is checked.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box.

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>Toggle</i>	Toggles the check box state.
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>Uncheck</i>	Unchecks the check box.
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using

Name	Description
WaitForDisappearance	the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTCheckboxMenuItem Class

Description

Represents a check box control that can be inserted in a menu item.

Inheritance Hierarchy

- [AWTMenuItem](#)
 - AWTCheckboxMenuItem

Syntax

```
'Declaration  
Public Class AWTCheckboxMenuItem _  
Inherits AWTMenuItem
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTMenuItem)
Checked	whether the check box of the menu item is checked.
Enabled	Whether the menu item is enabled. (Inherited from AWTMenuItem)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box of the menu item.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects the menu item. (Inherited from AWTMenuItem)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
Toggle	Toggles the check box state of the menu item.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks the check box of the menu item.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTChoice Class

Description

Represents a combo box.

Inheritance Hierarchy

- [AWTComponent](#)
 - AWTChoice

Syntax

```
'Declaration
Public Class AWTChoice _
Inherits AWTComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)

Name	Description
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
ItemCount	the number of items in the combo box.
Items	the list of combo box items.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	the index of the selected item. -1 (4Test: 0) is returned, if no item is currently selected.
SelectedItem	the selected item. NULL is returned, if no item is currently selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is

Name	Description
	passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItem	Returns the item at the specified index.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>Select</i>	Selects the specified item in the combo box.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTComponent Class

Description

AWTComponent is the base class for Java AWT.

Inheritance Hierarchy

- [TestObject](#)
 - AWTComponent
 - [AWTButton](#)
 - [AWTCanvas](#)

- [AWTCheckbox](#)
- [AWTChoice](#)
- [AWTContainer](#)
- [AWTLabel](#)
- [AWTList](#)
- [AWTRadioButton](#)
- [AWTScrollbar](#)
- [AWTTextComponent](#)

Syntax

```
'Declaration
Public Class AWTComponent _
Inherits TestObject _
Implements IClickable, IFocusable, IKeyable, INativeWindow
```

Properties

Name	Description
AccessibleName	The accessible name of the component.
AccessibleRole	The accessible role of the component.
Background	The background color of the component.
Cursor	The name of the cursor set in the component.
Enabled	whether the component is enabled.
Focusable	whether the component can be focused.
Font	The font of the component.
Foreground	The foreground color of the component.
Height	The current height of the component.
Name	The name of the component.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTContainer Class

Description

Represents a component that can contain other AWT components.

Inheritance Hierarchy

- [AWTComponent](#)
 - AWTContainer
 - [Applet](#)
 - [AppletContainer](#)
 - [AWTScrollPane](#)
 - [AWTWindow](#)
 - [JComponent](#)
 - [OracleFormsContainer](#)
 - [SplitPaneDivider](#)

Syntax

```
'Declaration  
Public Class AWTContainer _  
Inherits AWTComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTDialog Class

Description

Represents a dialog box.

Inheritance Hierarchy

- [AWTWindow](#)
 - AWTDialog
 - [JDialog](#)

Syntax

```
'Declaration
Public Class AWTDialog _
Inherits AWTWindow
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Modal	whether the dialog is modal.
Name	The name of the component. (Inherited from AWTComponent)

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Resizable	whether the dialog is resizable by the user.
Text	The text of the control. (Inherited from TestObject)
Title	the title of the dialog.
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Maximize	Maximizes the window. (Inherited from IMoveable)

Name	Description
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTFrame Class

Description

Represents a top-level window with a title and a border.

Inheritance Hierarchy

- [AWTWindow](#)
 - AWTFrame
 - [JFrame](#)

Syntax

```
'Declaration  
Public Class AWTFrame _  
Inherits AWTWindow
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Resizable	whether the frame is resizable by the user.
Text	The text of the control. (Inherited from TestObject)
Title	the title of the frame.
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> ,

Name	Description
WaitForObject	because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTHorizontalScrollbar Class

Description

Represents a horizontal scroll bar.

Inheritance Hierarchy

- [AWTScrollbar](#)
 - AWTHorizontalScrollbar

Syntax

```
'Declaration
Public Class AWTHorizontalScrollbar _
Inherits AWTScrollbar
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)

Name	Description
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	the block increment value of the scroll bar. (Inherited from IAWTScroller)
Position	the current position of the scroll bar. (Inherited from IAWTScroller)
Range	the range value of the scroll bar. (Inherited from IAWTScroller)
StepSize	the unit increment value of the scroll bar. (Inherited from IAWTScroller)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>Page</i>	Increments the current scroll position by specified number of pages. (Inherited from <i>IAWTScroller</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>ScrollTo</i>	Scrolls to the specified position. (Inherited from <i>IAWTScroller</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>Step</i>	Increments the current scroll position by specified steps. (Inherited from <i>IAWTScroller</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTLabel Class

Description

Represents a class for placing text in a container.

Inheritance Hierarchy

- [AWTComponent](#)
 - AWTLabel

Syntax

```
'Declaration  
Public Class AWTLabel _  
Inherits AWTComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Alignment	the alignment of the label.
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTList Class

Description

Represents a list box.

Inheritance Hierarchy

- [AWTComponent](#)
 - AWTList

Syntax

```
'Declaration  
Public Class AWTList _  
Inherits AWTComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
ItemCount	the number of list items.
Items	a list of items in the list.
MultipleMode	whether the list allows multiple selection.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	the index of the currently selected item. -1 (4Test: 0) is returned, if no item is currently selected.

Name	Description
SelectedIndices	the indices of the currently selected items. An empty list is returned, if no items are currently selected.
SelectedItem	the currently selected list item. NULL is returned, if no item is currently selected.
SelectedItems	the currently selected list items. An empty list is returned, if no items are currently selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Deselect	Deselects the specified list item.
DeselectAll	Deselects all list items.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClickItem	Double-clicks a list item.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetItem</i>	Returns the list item at the specified index.
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the specified item.
SelectAll	Selects all items in the list.
SelectRange	Selects all items within the specified range.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By

Name	Description
	default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTMenu Class

Description

Represents a menu component that is accessed from a menu bar.

Inheritance Hierarchy

- [AWTMenuItem](#)
 - AWTMenu

Syntax

```
'Declaration
Public Class AWTMenu _
Inherits AWTMenuItem
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTMenuItem)
Enabled	Whether the menu item is enabled. (Inherited from AWTMenuItem)
Text	The text of the control. (Inherited from TestObject)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects the menu item. (Inherited from AWTMenuItem)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTMenuComponent Class

Description

Represents the base class of all menu-related controls.

Inheritance Hierarchy

- [TestObject](#)
 - AWTMenuComponent
 - [AWTMenuItem](#)

Syntax

```
'Declaration
Public Class AWTMenuComponent _
Inherits TestObject _
Implements IClickable, IFocusable, IKeyable
```

Properties

Name	Description
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTMenuItem Class

Description

Represents a menu item.

Inheritance Hierarchy

- [AWTMenuComponent](#)
 - AWTMenuItem
 - [AWTCheckboxMenuItem](#)
 - [AWTMenu](#)

Syntax

```
'Declaration
Public Class AWTMenuItem _
Inherits AWTMenuComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component.
Enabled	Whether the menu item is enabled.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects the menu item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTRadioButton Class

Description

Represents a radio button.

Inheritance Hierarchy

- [AWTComponent](#)
 - AWTRadioButton

Syntax

```
'Declaration
Public Class AWTRadioButton _
Inherits AWTComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Checked	Whether the radio button is checked.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)

Name	Description
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks the radio button.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)

Name	Description
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTScrollbar Class

Description

Represents a scroll bar. This class includes scroll bars that are parts of controls, such as the scroll bar on a list box.

Inheritance Hierarchy

- [AWTComponent](#)
 - AWTScrollbar
 - [AWTHorizontalScrollbar](#)
 - [AWTVerticalScrollbar](#)

Syntax

```
'Declaration
Public Class AWTScrollbar _
Inherits AWTComponent _
Implements IAWTScroller
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)

Name	Description
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	the block increment value of the scroll bar. (Inherited from IAWTScroller)
Position	the current position of the scroll bar. (Inherited from IAWTScroller)
Range	the range value of the scroll bar. (Inherited from IAWTScroller)
StepSize	the unit increment value of the scroll bar. (Inherited from IAWTScroller)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
Page	Increments the current scroll position by specified number of pages. (Inherited from IAWTScroller)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
ScrollTo	Scrolls to the specified position. (Inherited from IAWTScroller)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
Step	Increments the current scroll position by specified steps. (Inherited from IAWTScroller)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForProperty	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTScrollPane Class

Description

Represents a class that implements automatic horizontal or vertical scrolling for a single child component.

Inheritance Hierarchy

- [AWTContainer](#)
 - AWTScrollPane

Syntax

```
'Declaration  
Public Class AWTScrollPane _  
Inherits AWTContainer _  
Implements IAWTScrollable
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Position	the current scroll position of the scrollable component. (Inherited from IAWTScrollable)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Gets the horizontal scroll bar. (Inherited from IAWTScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Gets the vertical scroll bar. (Inherited from IAWTScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
ScrollTo	Scrolls to the specified position. (Inherited from IAWTScrollable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTTextArea Class

Description

Represents a text area.

Inheritance Hierarchy

- [AWTTextComponent](#)
 - AWTTextArea

Syntax

```
'Declaration  
Public Class AWTTextArea _  
Inherits AWTTextComponent _  
Implements IAWTScrollable
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
CaretPosition	the position of the text insertion caret for the text component. (Inherited from AWTTextComponent)
Columns	the number of columns in the text area.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Editable	whether the text of the text component can be changed by the user. (Inherited from AWTTextComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Position	the current scroll position of the scrollable component. (Inherited from IAWTScrollable)
Rows	The number of rows in the text area.
SelectedText	the current selected text. (Inherited from AWTTextComponent)
SelectionEnd	the end position of the selected text. (Inherited from AWTTextComponent)
SelectionStart	the start position of the selected text. (Inherited from AWTTextComponent)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all the text from the text component. (Inherited from AWTTextComponent)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Gets the horizontal scroll bar. (Inherited from IAWTScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Gets the vertical scroll bar. (Inherited from IAWTScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Insert	Inserts the specified text at the specified position.
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReplaceRange	Replaces text between the specified start and end positions with the specified replacement text.
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
ScrollTo	Scrolls to the specified position. (Inherited from IAWTScrollable)
SelectAll	Selects all the text in the text component. (Inherited from AWTTextComponent)
SetCaretPosition	Sets the position of the text insertion caret for this text component. (Inherited from AWTTextComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the text within the specified range. (Inherited from AWTTextComponent)
SetText	Sets the text of the text component to the specified text. (Inherited from AWTTextComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTTextComponent Class

Description

Represents the base class for the AWTTextField and AWTTextArea classes.

Inheritance Hierarchy

- [AWTComponent](#)
 - AWTTextComponent
 - [AWTTextArea](#)
 - [AWTTextField](#)

Syntax

```
'Declaration  
Public Class AWTTextComponent _  
Inherits AWTComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
CaretPosition	the position of the text insertion caret for the text component.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Editable	whether the text of the text component can be changed by the user.
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedText	the current selected text.
SelectionEnd	the end position of the selected text.
SelectionStart	the start position of the selected text.
Text	The text of the control. (Inherited from TestObject)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all the text from the text component.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SelectAll	Selects all the text in the text component.
SetCaretPosition	Sets the position of the text insertion caret for this text component.

Name	Description
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the text within the specified range.
SetText	Sets the text of the text component to the specified text.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by

Name	Description
WaitForPropertyNotEquals	the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTTextField Class

Description

Represents single-line and multi-line text field controls whose text can be modified by the user.

Inheritance Hierarchy

- [AWTTextComponent](#)
 - AWTTextField

Syntax

```
'Declaration
Public Class AWTTextField _
Inherits AWTTextComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
CaretPosition	the position of the text insertion caret for the text component. (Inherited from AWTTextComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)

Name	Description
EchoChar	the character that is displayed instead of the characters that are actually typed.
EchoCharSet	whether <i>EchoChar</i> is set to a value other than 0.
Editable	whether the text of the text component can be changed by the user. (Inherited from AWTTextComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
IsPassword	whether the text field is a password text field.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedText	the current selected text. (Inherited from AWTTextComponent)
SelectionEnd	the end position of the selected text. (Inherited from AWTTextComponent)
SelectionStart	the start position of the selected text. (Inherited from AWTTextComponent)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all the text from the text component. (Inherited from AWTTextComponent)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SelectAll	Selects all the text in the text component. (Inherited from AWTTextComponent)
SetCaretPosition	Sets the position of the text insertion caret for this text component. (Inherited from AWTTextComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the text within the specified range. (Inherited from AWTTextComponent)
SetText	Sets the text of the text component to the specified text. (Inherited from AWTTextComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTVerticalScrollbar Class

Description

Represents a vertical scroll bar.

Inheritance Hierarchy

- [AWTScrollbar](#)
 - AWTVerticalScrollbar

Syntax

```
'Declaration
Public Class AWTVerticalScrollbar _
Inherits AWTScrollbar
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	the block increment value of the scroll bar. (Inherited from IAWTScroller)

Name	Description
Position	the current position of the scroll bar. (Inherited from IAWTScroller)
Range	the range value of the scroll bar. (Inherited from IAWTScroller)
StepSize	the unit increment value of the scroll bar. (Inherited from IAWTScroller)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
Page	Increments the current scroll position by specified number of pages. (Inherited from IAWTScroller)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
ScrollTo	Scrolls to the specified position. (Inherited from IAWTScroller)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
Step	Increments the current scroll position by specified steps. (Inherited from IAWTScroller)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

AWTWindow Class

Description

Represents a top-level window with no borders and no menubar.

Inheritance Hierarchy

- [AWTContainer](#)
 - AWTWindow
 - [AWTDialog](#)
 - [AWTFrame](#)
 - [JWindow](#)

Syntax

```
'Declaration
Public Class AWTWindow _
Inherits AWTContainer _
Implements IMoveable
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)

Name	Description
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

BasicArrowButton Class

Description

Represents a push button with an arrow in one of the cardinal directions.

Inheritance Hierarchy

- [JButton](#)
 - BasicArrowButton

Syntax

```
'Declaration
Public Class BasicArrowButton _
Inherits JButton
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Direction	the direction the arrow points to.
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)

Name	Description
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button. (Inherited from AbstractButton)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

IAWTScrollable Interface

Description

Represents an interface for AWT objects that can have two scroll bars.

Inheritance Hierarchy

- [IBaseScrollable](#)
 - IAWTScrollable

Syntax

```
'Declaration
Public Class IAWTScrollable _
Inherits IBaseScrollable
```

Properties

Name	Description
Position	the current scroll position of the scrollable component. (Inherited from IBaseScrollable)

Methods

Name	Description
GetHorizontalBar	Gets the horizontal scroll bar.
GetVerticalBar	Gets the vertical scroll bar.

Name	Description
ScrollTo	Scrolls to the specified position. (Inherited from IBaseScrollable)

IAWTScroller Interface

Description

Represents an interface for scrollable AWT objects.

Syntax

```
'Declaration
Public Class IAWTScroller
```

Properties

Name	Description
PageSize	the block increment value of the scroll bar.
Position	the current position of the scroll bar.
Range	the range value of the scroll bar.
StepSize	the unit increment value of the scroll bar.

Methods

Name	Description
Page	Increments the current scroll position by specified number of pages.
ScrollTo	Scrolls to the specified position.
Step	Increments the current scroll position by specified steps.

IBaseScrollable Interface

Description

Represents an interface for other scrollable controls to avoid duplication.

Syntax

```
'Declaration
Public Class IBaseScrollable
```

Properties

Name	Description
Position	the current scroll position of the scrollable component.

Methods

Name	Description
ScrollTo	Scrolls to the specified position.

IOracleFormsMenuBase Interface

Description

Represents a control that defines choices for users to select.

Syntax

```
'Declaration  
Public Class IOracleFormsMenuBase
```

Properties

Name	Description
Checked	Whether the MenuItem is checked.

IOracleFormsScrollable Interface

Description

Represents an interface for scrollable oracle forms objects.

Syntax

```
'Declaration  
Public Class IOracleFormsScrollable
```

Properties

Name	Description
HorizontalScrollbarPageSize	the block increment value of the scroll bar.
HorizontalScrollbarPosition	the current position of the scroll bar.
HorizontalScrollbarRange	the range value of the scroll bar.
HorizontalScrollbarStepSize	the unit increment value of the scroll bar.
VerticalScrollbarPageSize	the block increment value of the scroll bar.
VerticalScrollbarPosition	the current position of the scroll bar.
VerticalScrollbarRange	the range value of the scroll bar.
VerticalScrollbarStepSize	the unit increment value of the scroll bar.

Methods

Name	Description
GetHorizontalScrollbar	Gets the horizontal scroll bar.
GetVerticalScrollbar	Gets the vertical scroll bar.

Name	Description
HorizontalScrollbarPage	Increments the current scroll position by specified number of pages.
HorizontalScrollbarScrollTo	Scrolls to the specified position.
HorizontalScrollbarScrollToMax	Scrolls to the maximum position of the scroller.
HorizontalScrollbarScrollToMin	Scrolls to the minimum position of the scroller.
HorizontalScrollbarStep	Increments the current scroll position by specified steps.
VerticalScrollbarPage	Increments the current scroll position by specified number of pages.
VerticalScrollbarScrollTo	Scrolls to the specified position.
VerticalScrollbarScrollToMax	Scrolls to the maximum position of the scroller.
VerticalScrollbarScrollToMin	Scrolls to the minimum position of the scroller.
VerticalScrollbarStep	Increments the current scroll position by specified steps.

IOracleFormsScroller Interface

Description

Represents an interface for scrollable oracle forms objects.

Inheritance Hierarchy

- [IAWTScroller](#)
 - IOracleFormsScroller

Syntax

```
'Declaration
Public Class IOracleFormsScroller _
Inherits IAWTScroller
```

Properties

Name	Description
PageSize	the block increment value of the scroll bar. (Inherited from IAWTScroller)
Position	the current position of the scroll bar. (Inherited from IAWTScroller)
Range	the range value of the scroll bar. (Inherited from IAWTScroller)
StepSize	the unit increment value of the scroll bar. (Inherited from IAWTScroller)

Methods

Name	Description
Page	Increments the current scroll position by specified number of pages. (Inherited from IAWTScroller)
ScrollTo	Scrolls to the specified position. (Inherited from IAWTScroller)
ScrollToMax	Scrolls to the maximum position of the scroller.
ScrollToMin	Scrolls to the minimum position of the scroller.
Step	Increments the current scroll position by specified steps. (Inherited from IAWTScroller)

JButton Class

Description

Represents a push button.

Inheritance Hierarchy

- [AbstractButton](#)
 - JButton
 - [BasicArrowButton](#)

Syntax

```
'Declaration  
Public Class JButton _  
Inherits AbstractButton
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)

Name	Description
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button. (Inherited from AbstractButton)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JCheckBox Class

Description

Represents a check box.

Inheritance Hierarchy

- [JToggleButton](#)
 - JCheckBox

Syntax

```
'Declaration
Public Class JCheckBox _
Inherits JToggleButton
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)

Name	Description
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Selected	whether the toggle button is selected. (Inherited from JToggleButton)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button. (Inherited from AbstractButton)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
Toggle	Toggles the button. (Inherited from JToggleButton)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks the check box.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JCheckBoxMenuItem Class

Description

Represents a check box control that can be included in a menu item.

Inheritance Hierarchy

- [JMenuItem](#)
 - JCheckBoxMenuItem

Syntax

```
'Declaration  
Public Class JCheckBoxMenuItem _  
Inherits JMenuItem
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Checked	whether the check box in a menu item is checked.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The

Name	Description
	<p>captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button. (Inherited from AbstractButton)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JColorChooser Class

Description

Represents a color chooser.

Inheritance Hierarchy

- [JComponent](#)
 - JColorChooser

Syntax

```
'Declaration  
Public Class JColorChooser _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Color	the current color value from the color chooser.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetColor	Sets the current color of the color chooser to the specified color.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)

Name	Description
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JComboBox Class

Description

Represents a combo box.

Inheritance Hierarchy

- [JComponent](#)
 - JComboBox

Syntax

```
'Declaration
Public Class JComboBox _
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Editable	whether the combo box is editable.
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
ItemCount	the number of items in the list.

Name	Description
Items	the list of all contained items.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	the index of the currently selected item.
SelectedItem	the current selected item.
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItem	Returns the item at the specified index.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the specified item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JComponent Class

Description

Represents a class for all components except top-level containers.

Inheritance Hierarchy

- [AWTContainer](#)
 - JComponent
 - [AbstractButton](#)
 - [JColorChooser](#)
 - [JComboBox](#)
 - [JLabel](#)
 - [JLayeredPane](#)
 - [JList](#)
 - [JMenuBar](#)
 - [JPanel](#)
 - [JPopupMenu](#)
 - [JProgressBar](#)
 - [JRootPane](#)
 - [JScrollBar](#)
 - [JScrollPane](#)
 - [JSlider](#)
 - [JSpinner](#)
 - [JSplitPane](#)

- [JTabbedPane](#)
- [JTable](#)
- [JTableHeader](#)
- [JTextComponent](#)
- [JToolBar](#)
- [JTree](#)
- [JViewport](#)

Syntax

```
'Declaration
Public Class JComponent _
Inherits AWTContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component.
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JDesktopPane Class

Description

Represents a desktop pane.

Inheritance Hierarchy

- [JLayeredPane](#)
 - JDesktopPane

Syntax

```
'Declaration  
Public Class JDesktopPane _  
Inherits JLayeredPane
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JDialog Class

Description

Represents a dialog box.

Inheritance Hierarchy

- [AWTDialog](#)
 - JDialog

Syntax

```
'Declaration
Public Class JDialog _
Inherits AWTDialog
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Modal	whether the dialog is modal. (Inherited from AWTDialog)
Name	The name of the component. (Inherited from AWTComponent)

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Resizable	whether the dialog is resizable by the user. (Inherited from AWTDialog)
Text	The text of the control. (Inherited from TestObject)
Title	the title of the dialog. (Inherited from AWTDialog)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Maximize	Maximizes the window. (Inherited from IMoveable)

Name	Description
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JEditorPane Class

Description

Represents a text component where you can edit different types of content.

Inheritance Hierarchy

- [JTextComponent](#)
 - JEditorPane
 - [JTextPane](#)

Syntax

```
'Declaration  
Public Class JEditorPane _  
Inherits JTextComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
CaretPosition	the position of the text insertion caret for the text component. (Inherited from JTextComponent)
ContentType	the type of content that this editor is currently set to deal with.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Editable	whether the text of the text component can be changed by the user. (Inherited from JTextComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
FormattedText	the text of the editor pane.
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Page	the current URL being displayed.
SelectedText	the current selected text. (Inherited from JTextComponent)

Name	Description
SelectionEnd	the end position of the selection. (Inherited from JTextComponent)
SelectionStart	the start position of the selection. (Inherited from JTextComponent)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all the text from text component. (Inherited from JTextComponent)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject . (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from AWTComponent)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SelectAll	Selects all the content in the text component. (Inherited from JTextComponent)
SetCaretPosition	Sets the position of the text insertion caret for this text component. (Inherited from JTextComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the text within the specified range. (Inherited from JTextComponent)
SetText	Sets the text content to the specified text. (Inherited from JTextComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JFrame Class

Description

Represents a class for an extended version of `AWTFrame` that adds support for Java Swing.

Inheritance Hierarchy

- [AWTFrame](#)
 - JFrame

Syntax

```
'Declaration
Public Class JFrame _
Inherits AWTFrame
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)

Name	Description
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Resizable	whether the frame is resizable by the user. (Inherited from AWTFrame)
Text	The text of the control. (Inherited from TestObject)
Title	the title of the frame. (Inherited from AWTFrame)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 Size 	Resizes the window. (Inherited from IMoveable)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JHorizontalScrollBar Class

Description

Represents a horizontal scroll bar.

Inheritance Hierarchy

- [JScrollBar](#)
 - JHorizontalScrollBar

Syntax

```
'Declaration
Public Class JHorizontalScrollBar _
Inherits JScrollBar
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)

Name	Description
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	the block increment value of the scroll bar. (Inherited from IAWTScroller)
Position	the current position of the scroll bar. (Inherited from IAWTScroller)
Range	the range value of the scroll bar. (Inherited from IAWTScroller)
StepSize	the unit increment value of the scroll bar. (Inherited from IAWTScroller)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
Page	Increments the current scroll position by specified number of pages. (Inherited from IAWTScroller)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
ScrollTo	Scrolls to the specified position. (Inherited from IAWTScroller)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
Step	Increments the current scroll position by specified steps. (Inherited from IAWTScroller)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
WaitForObject	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

JLabel Class

Description

Represents a display area for displaying a short text, image, or both.

Inheritance Hierarchy

- [JComponent](#)
 - JLabel

Syntax

```
'Declaration
Public Class JLabel _
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Alignment	the alignment of the label.
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForProperty	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JLayeredPane Class

Description

Represents a layered pane.

Inheritance Hierarchy

- [JComponent](#)
 - JLayeredPane

- [JDesktopPane](#)

Syntax

```
'Declaration
Public Class JLayeredPane _
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JList Class

Description

Represents a list box.

Inheritance Hierarchy

- [JComponent](#)
 - JList

Syntax

```
'Declaration  
Public Class JList _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
ItemCount	the number of list items.
Items	the list of all contained list items.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	the index of the selected list item.
SelectedIndices	the list of indices of the selected list items.
SelectedItem	the name or index of the selected list item.
SelectedItems	the list of names or indices of the selected list items.
Text	The text of the control. (Inherited from TextObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
ClickItem	Clicks the specified list item.
Deselect	Deselects the specified list item.
DeselectAll	Deselects all list items.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClickItem	Double-clicks a list item.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
FindItem	Returns the index of an item in the list.
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
<i>GetItem</i>	Returns the item at the specified index.
<i>GetItemRect</i>	Returns the size and position of the item relative to the list.
<i>GetItemValue</i>	Returns the value object of the list item.
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
ScrollItemIntoView	Scrolls an item into view.
Select	Selects a list item.
SelectAll	Selects all list elements.
SelectRange	Selects a specified range of list elements.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> ,

Name	Description
WaitForObject	because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JMenu Class

Description

Represents a menu. A menu is a menu item that has child items.

Inheritance Hierarchy

- [JMenuItem](#)
 - JMenu

Syntax

```
'Declaration
Public Class JMenu _
Inherits JMenuItem
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)

Name	Description
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)

Name	Description
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button. (Inherited from AbstractButton)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JMenuBar Class

Description

Represents a menu bar.

Inheritance Hierarchy

- [JComponent](#)
 - JMenuBar

Syntax

```
'Declaration
Public Class JMenuBar _
Inherits JComponent
```


Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JMenuItem Class

Description

Represents a menu item.

Inheritance Hierarchy

- [AbstractButton](#)
 - JMenuItem
 - [JCheckBoxMenuItem](#)
 - [JMenu](#)

- [JRadioButtonMenuItem](#)

Syntax

```
'Declaration
Public Class JMenuItem _
Inherits AbstractButton
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button. (Inherited from AbstractButton)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JPanel Class

Description

Represents a container class for lightweight components.

Inheritance Hierarchy

- [JComponent](#)
 - JPanel

Syntax

```
'Declaration  
Public Class JPanel _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JPasswordField Class

Description

Represents a single-line field that does not show the original characters.

Inheritance Hierarchy

- [JTextField](#)
 - JPasswordField

Syntax

```
'Declaration
Public Class JPasswordField _
Inherits JTextField
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Alignment	the alignment of the text field. (Inherited from JTextField)
Background	The background color of the component. (Inherited from AWTComponent)
CaretPosition	the position of the text insertion caret for the text component. (Inherited from JTextComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
EchoChar	the character that is displayed instead of the characters that are actually typed.
EchoCharSet	whether EchoChar is set to a value other than 0.
Editable	whether the text of the text component can be changed by the user. (Inherited from JTextComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)

Name	Description
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedText	the current selected text. (Inherited from JTextComponent)
SelectionEnd	the end position of the selection. (Inherited from JTextComponent)
SelectionStart	the start position of the selection. (Inherited from JTextComponent)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all the text from text component. (Inherited from JTextComponent)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from TestObject)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SelectAll	Selects all the content in the text component. (Inherited from JTextComponent)
SetCaretPosition	Sets the position of the text insertion caret for this text component. (Inherited from JTextComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the text within the specified range. (Inherited from JTextComponent)
SetText	Sets the text content to the specified text. (Inherited from JTextComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForProperty	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JPopupMenu Class

Description

Represents a context menu.

Inheritance Hierarchy

- [JComponent](#)
 - JPopupMenu

Syntax

```
'Declaration  
Public Class JPopupMenu _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel

Name	Description
	must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JProgressBar Class

Description

Displays the progress of an operation.

Inheritance Hierarchy

- [JComponent](#)
 - JProgressBar

Syntax

```
'Declaration  
Public Class JProgressBar _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Indeterminate	whether the progress bar is indeterminate.
Maximum	the progress bar's maximum value.
Minimum	the progress bar's minimum value.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
ProgressString	the string representation of the current progress.
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JRadioButton Class

Description

Represents a radio button.

Inheritance Hierarchy

- [JToggleButton](#)
 - JRadioButton

Syntax

```
'Declaration
Public Class JRadioButton _
Inherits JToggleButton
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)

Name	Description
Selected	whether the toggle button is selected. (Inherited from JToggleButton)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks the radio button.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)

Name	Description
Select	Selects the button. (Inherited from AbstractButton)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
Toggle	Toggles the button. (Inherited from JToggleButton)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by

Name	Description
WaitForPropertyNotEquals	the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JRadioButtonMenuItem Class

Description

Represents a radio button menu item.

Inheritance Hierarchy

- [JMenuItem](#)
 - JRadioButtonMenuItem

Syntax

```
'Declaration
Public Class JRadioButtonMenuItem _
Inherits JMenuItem
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Checked	whether the radio button in a menu item is checked.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)

Name	Description
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button. (Inherited from AbstractButton)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when

Name	Description
	a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JRootPane Class

Description

Represents a root pane.

Inheritance Hierarchy

- [JComponent](#)
 - JRootPane

Syntax

```
'Declaration
Public Class JRootPane _
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)

Name	Description
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JScrollBar Class

Description

Represents a scroll bar.

Inheritance Hierarchy

- [JComponent](#)
 - JScrollBar
 - [JHorizontalScrollBar](#)
 - [JVerticalScrollBar](#)

Syntax

```
'Declaration
Public Class JScrollBar _
Inherits JComponent _
Implements IAWTScroller
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	the block increment value of the scroll bar. (Inherited from IAWTScroller)
Position	the current position of the scroll bar. (Inherited from IAWTScroller)
Range	the range value of the scroll bar. (Inherited from IAWTScroller)
StepSize	the unit increment value of the scroll bar. (Inherited from IAWTScroller)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default

Name	Description
	<p>bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>Page</i>	Increments the current scroll position by specified number of pages. (Inherited from <i>IAWTScroller</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>ScrollTo</i>	Scrolls to the specified position. (Inherited from <i>IAWTScroller</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>Step</i>	Increments the current scroll position by specified steps. (Inherited from <i>IAWTScroller</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JScrollPane Class

Description

Represents a scroll pane which allows scrolling its content which may be larger than the pane itself.

Inheritance Hierarchy

- [JComponent](#)
 - JScrollPane

Syntax

```
'Declaration
Public Class JScrollPane _
Inherits JComponent _
Implements IBaseScrollable
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Position	the current scroll position of the scrollable component. (Inherited from IBaseScrollable)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetHorizontalBar	Gets the horizontal scroll bar.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Gets the vertical scroll bar.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject . (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
ScrollTo	Scrolls to the specified position. (Inherited from IBaseScrollable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JSlider Class

Description

Represents a slider control which can select a value within a range.

Inheritance Hierarchy

- [JComponent](#)
 - JSlider

Syntax

```
'Declaration
Public Class JSlider _
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)

Name	Description
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Inverted	the value is <i>true</i> if the value-range shown for the slider is reversed.
Maximum	the maximum value supported by the slider.
Minimum	the minimum value supported by the slider.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Position	the position of the slider.
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetPosition</i>	Sets the current position of the slider.
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JSpinner Class

Description

Represents a spinner control which can iterate a sequence of values.

Inheritance Hierarchy

- [JComponent](#)
 - JSpinner

Syntax

```
'Declaration  
Public Class JSpinner _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel

Name	Description
	must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Next	Sets the spinner to the object in the sequence that comes after the object returned by the Value property.
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Previous	Sets the spinner to the object in the sequence that comes before the object returned by the Value property.
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JSplitPane Class

Description

Represents a split pane control which divides two components horizontally or vertically.

Inheritance Hierarchy

- [JComponent](#)
 - JSplitPane

Syntax

```
'Declaration  
Public Class JSplitPane _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
DividerLocation	the current divider location.
DividerSize	the size of the divider.
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
MaximumDividerLocation	the maximum location of the divider.
MinimumDividerLocation	the minimum location of the divider.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)

Name	Description
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
SetPosition	Sets the position of the divider.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JTabbedPane Class

Description

Represents a tabbed pane.

Inheritance Hierarchy

- [JComponent](#)
 - JTabbedPane

Syntax

```
'Declaration
Public Class JTabbedPane _
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)

Name	Description
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedTab	the currently selected tab.
SelectedTabIndex	the index of the currently selected tab or -1 (4Test: 0) if no tab is selected.
TabCount	the number of tabs.
TabRunCount	the number of tab runs currently used to display the tabs.
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetTabIndex	Returns the tab index.
GetTabTitle	Returns the tab title.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
IsTabEnabled	Returns whether or not the tab is currently enabled.

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the specified tab.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JTable Class

Description

Represents a table control which has rows and columns.

Inheritance Hierarchy

- [JComponent](#)
 - JTable

Syntax

```
'Declaration
Public Class JTable _
Inherits JComponent
```


Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
AllCellTexts	the text values of all cells in the table. The cells are grouped in rows, where each row contains the texts of its cells.
AllCellValues	the values of all cells in the table. The cells are grouped in rows, where each row contains the values of its cells.
Background	The background color of the component. (Inherited from AWTComponent)
ColumnCount	the number of columns in the table.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
EditingColumn	the name or index of the column that contains the cell currently being edited.
EditingRow	the index of the row that contains the cell currently being edited.
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
RowCount	the number of rows that can be shown in the table.
SelectedColumn	the index or name of the selected column. -1 (4Test: 0) is returned if no column is selected.
SelectedColumns	the list of indices of all selected columns.
SelectedRow	the index of the selected row, -1 (4Test: 0) is returned if no row is selected.
SelectedRows	the list of indices of all selected rows.
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)

Name	Description
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ClearSelection	Deselects all selected columns and rows.
Click	Clicks on the object. (Inherited from IClickable)
ClickCell	Clicks the specified cell in the table.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClickCell	Double-clicks the specified cell in the table.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
FindCell	Returns the cell containing the specified text.
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCellText	Returns the cell text.
GetCellValue	Returns the cell value.
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetColumnName	Returns the name of the column at the specified index.

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsCellEditable</i>	Returns true if the specified cell is editable.
<i>IsCellSelected</i>	Returns true if the specified cell is selected; otherwise false.
<i>IsColumnSelected</i>	Returns true if the specified column is selected.
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>IsRowSelected</i>	Returns true if the specified row is selected.
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MoveColumn</i>	Moves the column to the position of the target column.
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ResizeColumn	Resizes the specified column.
ScrollCellIntoView	Scrolls the specified cell into view.
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SelectAll	Selects all rows, columns, and cells in the table.
SelectColumn	Selects the specified column.
SelectRow	Selects the specified row.
SetCellValue	Sets the specified value for the specified cell in the table.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JTableHeader Class

Description

Represents table header class for the table component.

Inheritance Hierarchy

- [JComponent](#)
 - `JTableHeader`

Syntax

```
'Declaration  
Public Class JTableHeader _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel

Name	Description
	must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
ClickColumnHeader	Clicks the column header of the table.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JTextArea Class

Description

Represents a multiple-line area control that displays plain text.

Inheritance Hierarchy

- [JTextComponent](#)
 - JTextArea

Syntax

```
'Declaration  
Public Class JTextArea _  
Inherits JTextComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
CaretPosition	the position of the text insertion caret for the text component. (Inherited from JTextComponent)
Columns	the number of columns in the text area.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Editable	whether the text of the text component can be changed by the user. (Inherited from JTextComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Rows	The number of rows in the text area.
SelectedText	the current selected text. (Inherited from JTextComponent)
SelectionEnd	the end position of the selection. (Inherited from JTextComponent)

Name	Description
SelectionStart	the start position of the selection. (Inherited from JTextComponent)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all the text from text component. (Inherited from JTextComponent)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Insert</i>	Inserts the specified text at the specified position.
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReplaceRange	Replaces text within the specified range with the new text specified.
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SelectAll	Selects all the content in the text component. (Inherited from JTextComponent)
SetCaretPosition	Sets the position of the text insertion caret for this text component. (Inherited from JTextComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the text within the specified range. (Inherited from JTextComponent)
SetText	Sets the text content to the specified text. (Inherited from JTextComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JTextComponent Class

Description

Represents a text control that can be marked with attributes that are represented graphically.

Inheritance Hierarchy

- [JComponent](#)
 - JTextComponent
 - [JEditorPane](#)
 - [JTextArea](#)
 - [JTextField](#)

Syntax

```
'Declaration
Public Class JTextComponent _
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
CaretPosition	the position of the text insertion caret for the text component.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Editable	whether the text of the text component can be changed by the user.
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedText	the current selected text.
SelectionEnd	the end position of the selection.
SelectionStart	the start position of the selection.
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default

Name	Description
	<p>bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
Clear	Removes all the text from text component.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	<p>Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)</p>
Exists	<p>Checks if an object exists in the application under test. (Inherited from TestObject)</p>
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SelectAll	Selects all the content in the text component.
SetCaretPosition	Sets the position of the text insertion caret for this text component.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the text within the specified range.
SetText	Sets the text content to the specified text.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JTextField Class

Description

Represents a single-line text field control whose text can be modified by the user.

Inheritance Hierarchy

- [JTextComponent](#)
 - JTextField
 - [JPasswordField](#)

Syntax

```
'Declaration
Public Class JTextField _
Inherits JTextComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Alignment	the alignment of the text field.
Background	The background color of the component. (Inherited from AWTComponent)
CaretPosition	the position of the text insertion caret for the text component. (Inherited from JTextComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Editable	whether the text of the text component can be changed by the user. (Inherited from JTextComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)

Name	Description
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedText	the current selected text. (Inherited from JTextComponent)
SelectionEnd	the end position of the selection. (Inherited from JTextComponent)
SelectionStart	the start position of the selection. (Inherited from JTextComponent)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all the text from text component. (Inherited from JTextComponent)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SelectAll	Selects all the content in the text component. (Inherited from JTextComponent)
SetCaretPosition	Sets the position of the text insertion caret for this text component. (Inherited from JTextComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the text within the specified range. (Inherited from JTextComponent)
SetText	Sets the text content to the specified text. (Inherited from JTextComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JTextPane Class

Description

Represents a text pane control that can be marked with attributes that are represented graphically.

Inheritance Hierarchy

- [JEditorPane](#)
 - JTextPane

Syntax

```
'Declaration  
Public Class JTextPane _  
Inherits JEditorPane
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
CaretPosition	the position of the text insertion caret for the text component. (Inherited from JTextComponent)
CharacterAttributes	the character attributes in effect at the location of the caret.
ContentType	the type of content that this editor is currently set to deal with. (Inherited from JEditorPane)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Editable	whether the text of the text component can be changed by the user. (Inherited from JTextComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
FormattedText	the text of the editor pane. (Inherited from JEditorPane)
Height	The current height of the component. (Inherited from AWTComponent)
LogicalStyle	the logical style assigned to the current paragraph in effect at the location of the caret.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Page	the current URL being displayed. (Inherited from JEditorPane)
ParagraphAttributes	the attributes of the current paragraph in effect at the location of the caret.
SelectedText	the current selected text. (Inherited from JTextComponent)

Name	Description
SelectionEnd	the end position of the selection. (Inherited from JTextComponent)
SelectionStart	the start position of the selection. (Inherited from JTextComponent)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all the text from text component. (Inherited from JTextComponent)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SelectAll	Selects all the content in the text component. (Inherited from JTextComponent)
SetCaretPosition	Sets the position of the text insertion caret for this text component. (Inherited from JTextComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the text within the specified range. (Inherited from JTextComponent)
SetText	Sets the text content to the specified text. (Inherited from JTextComponent)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option

Name	Description
	<i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JToggleButton Class

Description

Represents a two-state button.

Inheritance Hierarchy

- [AbstractButton](#)
 - JToggleButton
 - [JCheckBox](#)
 - [JRadioButton](#)

Syntax

```
'Declaration
Public Class JToggleButton _
Inherits AbstractButton
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Selected	whether the toggle button is selected.
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant

Name	Description
	<i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button. (Inherited from AbstractButton)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
Toggle	Toggles the button.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JToolBar Class

Description

Represents a toolbar.

Inheritance Hierarchy

- [JComponent](#)
 - JToolBar

Syntax

```
'Declaration  
Public Class JToolBar _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JTree Class

Description

Represents a tree.

Inheritance Hierarchy

- [JComponent](#)
 - JTree

Syntax

```
'Declaration  
Public Class JTree _  
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
IsEditable	whether the tree is editable.
IsEditing	whether the tree is being edited.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedItemPath	the selected item path.
SelectedItemPaths	the list of selected item paths.

Name	Description
SelectedItemText	the selected item text.
SelectionCount	the number of tree nodes selected.
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
VisibleItemPaths	the visible items in the tree as item paths.
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CancelEditing	Cancels the current editing session.
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ClearSelection	Clears the selection.
Click	Clicks on the object. (Inherited from IClickable)
ClickNode	Clicks on a tree node specified by the item path.
Collapse	Collapses the tree node identified by the specified item path.
CollapseAll	Collapses all tree nodes.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClickNode	Double-clicks on a tree node specified by the item path.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Expand	Expands the tree node identified by the specified item path.
ExpandAll	Expands all tree nodes.

Name	Description
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetItemRect</i>	Returns the size and position of the item relative to the tree.
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsExpanded</i>	Returns 'true' if the specified item path is expanded.
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>IsPathEditable</i>	whether an item path is editable.

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
ScrollPathIntoView	Scrolls the given item path to a visible position.
Select	Selects a tree node identified by the specified item path.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartEditing	Starts editing at given item path or current selection.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopEditing	Stops the current editing session.
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the

Name	Description
WaitForDisappearance	option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JVerticalScrollBar Class

Description

Represents a vertical scroll bar.

Inheritance Hierarchy

- [JScrollBar](#)
 - JVerticalScrollBar

Syntax

```
'Declaration  
Public Class JVerticalScrollBar _  
Inherits JScrollBar
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	the block increment value of the scroll bar. (Inherited from IAWTScroller)
Position	the current position of the scroll bar. (Inherited from IAWTScroller)
Range	the range value of the scroll bar. (Inherited from IAWTScroller)
StepSize	the unit increment value of the scroll bar. (Inherited from IAWTScroller)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>Page</i>	Increments the current scroll position by specified number of pages. (Inherited from <i>IAWTScroller</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>ScrollTo</i>	Scrolls to the specified position. (Inherited from <i>IAWTScroller</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>Step</i>	Increments the current scroll position by specified steps. (Inherited from <i>IAWTScroller</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JViewport Class

Description

Represents the Swing viewport.

Inheritance Hierarchy

- [JComponent](#)
 - JViewport

Syntax

```
'Declaration
Public Class JViewport _
Inherits JComponent
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
ToolTipText	the tool tip text of a Swing component. (Inherited from JComponent)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
ViewRectangle	the visible part of the view, in view coordinates.
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)

Name	Description
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

JWindow Class

Description

Represents a container that can be displayed anywhere on the user's desktop.

Inheritance Hierarchy

- [AWTWindow](#)
 - JWindow

Syntax

```
'Declaration
Public Class JWindow _
Inherits AWTWindow
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)

Name	Description
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)

Name	Description
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsApplication Class

Description

Represents an Oracle Forms application.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsApplication

Syntax

```
'Declaration  
Public Class OracleFormsApplication _  
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SplashScreenRunning	whether the application currently displays a splash screen.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsButton Class

Description

Represents a push button.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsButton

Syntax

```
'Declaration  
Public Class OracleFormsButton _  
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Default	Whether the button is the default button.
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the button.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsCheckbox Class

Description

Represents a check box.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsCheckbox

Syntax

```
'Declaration
Public Class OracleFormsCheckbox _
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Checked	whether the check box is checked.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)

Name	Description
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from AWTComponent)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)

Name	Description
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks the check box.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An

Name	Description
WaitForPropertyNotEquals	exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsChoice Class

Description

Represents a choice.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsChoice
 - [OracleFormsComboBox](#)
 - [OracleFormsListBox](#)
 - [OracleFormsPopList](#)

Syntax

```
'Declaration
Public Class OracleFormsChoice _
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)

Name	Description
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
ItemCount	the number of items in the choice.
Items	the list of choice items.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	the index of the selected item. -1 (4Test: 0) is returned, if no item is currently selected.
SelectedItem	the selected item.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemText	Returns the item at the specified index.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the specified item in the choice.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF . Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsComboBox Class

Description

Represents a combo box.

Inheritance Hierarchy

- [OracleFormsChoice](#)
 - OracleFormsComboBox

Syntax

```
'Declaration
Public Class OracleFormsComboBox _
Inherits OracleFormsChoice
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
DropDownOpen	whether the drop down is currently open.
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
ItemCount	the number of items in the choice. (Inherited from OracleFormsChoice)
Items	the list of choice items. (Inherited from OracleFormsChoice)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	the index of the selected item. -1 (4Test: 0) is returned, if no item is currently selected. (Inherited from OracleFormsChoice)
SelectedItem	the selected item. (Inherited from OracleFormsChoice)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemText	Returns the item at the specified index. (Inherited from OracleFormsChoice)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the specified item in the choice. (Inherited from OracleFormsChoice)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetText	Sets the text of the combo box to the specified text.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsContainer Class

Description

Represents a component that can contain other Oracle Forms components.

Inheritance Hierarchy

- [AWTContainer](#)
 - OracleFormsContainer
 - [OracleFormsApplication](#)
 - [OracleFormsButton](#)
 - [OracleFormsCheckbox](#)
 - [OracleFormsChoice](#)
 - [OracleFormsLabel](#)
 - [OracleFormsListView](#)
 - [OracleFormsMenu](#)
 - [OracleFormsMenuItem](#)
 - [OracleFormsRadioButton](#)
 - [OracleFormsScrollbar](#)
 - [OracleFormsStatusArea](#)
 - [OracleFormsStatusBar](#)
 - [OracleFormsStatusBarItem](#)
 - [OracleFormsStatusIndicator](#)
 - [OracleFormsTabBar](#)
 - [OracleFormsTabBarItem](#)
 - [OracleFormsTabPanel](#)
 - [OracleFormsTextField](#)
 - [OracleFormsTitleBar](#)
 - [OracleFormsToolBar](#)
 - [OracleFormsToolBarItem](#)
 - [OracleFormsTree](#)

Syntax

```
'Declaration
Public Class OracleFormsContainer _
Inherits AWTContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)

Name	Description
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change

Name	Description
	the timeout by setting the value of the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsHorizontalScrollbar Class

Description

Represents a horizontal scroll bar.

Inheritance Hierarchy

- [OracleFormsScrollbar](#)
 - OracleFormsHorizontalScrollbar

Syntax

```
'Declaration
Public Class OracleFormsHorizontalScrollbar _
Inherits OracleFormsScrollbar
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)

Name	Description
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScroller)
Position	the current position of the scroll bar. (Inherited from IOracleFormsScroller)
Range	the range value of the scroll bar. (Inherited from IOracleFormsScroller)
StepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScroller)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the

Name	Description
	constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>Page</i>	Increments the current scroll position by specified number of pages. (Inherited from <i>IOracleFormsScroller</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>ScrollTo</i>	Scrolls to the specified position. (Inherited from <i>IOracleFormsScroller</i>)
<i>ScrollToMax</i>	Scrolls to the maximum position of the scroller. (Inherited from <i>IOracleFormsScroller</i>)
<i>ScrollToMin</i>	Scrolls to the minimum position of the scroller. (Inherited from <i>IOracleFormsScroller</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>Step</i>	Increments the current scroll position by specified steps. (Inherited from <i>IOracleFormsScroller</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsLabel Class

Description

Represents a class for placing text in a container.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsLabel

Syntax

'Declaration

Public Class OracleFormsLabel _

Inherits OracleFormsContainer

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)

Name	Description
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

OracleFormsListBox Class

Description

Represents a ListBox.

Inheritance Hierarchy

- [OracleFormsChoice](#)
 - OracleFormsListBox

Syntax

```
'Declaration
Public Class OracleFormsListBox _
Inherits OracleFormsChoice _
Implements IOracleFormsScrollable
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)

Name	Description
Height	The current height of the component. (Inherited from AWTComponent)
HorizontalScrollbarPageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarPosition	the current position of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarRange	the range value of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarStepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
ItemCount	the number of items in the choice. (Inherited from OracleFormsChoice)
Items	the list of choice items. (Inherited from OracleFormsChoice)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	the index of the selected item. -1 (4Test: 0) is returned, if no item is currently selected. (Inherited from OracleFormsChoice)
SelectedItem	the selected item. (Inherited from OracleFormsChoice)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
VerticalScrollbarPageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarPosition	the current position of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarRange	the range value of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarStepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
ClickItem	Clicks the specified list item.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClickItem	Double-clicks a list item.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalScrollbar	Gets the horizontal scroll bar. (Inherited from IOracleFormsScrollable)
GetItemText	Returns the item at the specified index. (Inherited from OracleFormsChoice)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalScrollbar	Gets the vertical scroll bar. (Inherited from IOracleFormsScrollable)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
HorizontalScrollbarPage	Increments the current scroll position by specified number of pages. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollTo	Scrolls to the specified position. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollToMax	Scrolls to the maximum position of the scroller. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollToMin	Scrolls to the minimum position of the scroller. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarStep	Increments the current scroll position by specified steps. (Inherited from IOracleFormsScrollable)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the specified item in the choice. (Inherited from OracleFormsChoice)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
VerticalScrollbarPage	Increments the current scroll position by specified number of pages. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollTo	Scrolls to the specified position. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollToMax	Scrolls to the maximum position of the scroller. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollToMin	Scrolls to the minimum position of the scroller. (Inherited from IOracleFormsScrollable)
VerticalScrollbarStep	Increments the current scroll position by specified steps. (Inherited from IOracleFormsScrollable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsListView Class

Description

Represents a listview.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsListView

Syntax

```
'Declaration
Public Class OracleFormsListView _
Inherits OracleFormsContainer _
Implements IOOracleFormsScrollable
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
AllCellTexts	the text values of all cells in the listview. The cells are grouped in rows, where each row contains the texts of its cells.
Background	The background color of the component. (Inherited from AWTComponent)
ColumnCount	the number of columns in the listview.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
FocusedRow	the index of the focused row, -1 (4Test: 0) is returned if no row is focused.
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
HorizontalScrollbarPageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarPosition	the current position of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarRange	the range value of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarStepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
RowCount	the number of rows in the listview.
SelectedRow	the index of the selected row, -1 (4Test: 0) is returned if no row is selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
VerticalScrollbarPageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarPosition	the current position of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarRange	the range value of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarStepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ClearSelection	Clears the selection.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCellText	Returns the cell text.
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetColumnName	Returns the name of the column at the specified index.

Name	Description
GetColumnWidth	Returns the width of the specified column.
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetHorizontalScrollbar	Gets the horizontal scroll bar. (Inherited from IOracleFormsScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalScrollbar	Gets the vertical scroll bar. (Inherited from IOracleFormsScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
HorizontalScrollbarPage	Increments the current scroll position by specified number of pages. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollTo	Scrolls to the specified position. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollToMax	Scrolls to the maximum position of the scroller. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollToMin	Scrolls to the minimum position of the scroller. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarStep	Increments the current scroll position by specified steps. (Inherited from IOracleFormsScrollable)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>IsRowVisible</i>	Returns whether the specified row is visible.
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>ScrollToRow</i>	Scrolls to the specified row.
<i>SelectRow</i>	Selects the specified row.
<i>SetColumnWidth</i>	Sets the width of the specified column.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
VerticalScrollbarPage	Increments the current scroll position by specified number of pages. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollTo	Scrolls to the specified position. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollToMax	Scrolls to the maximum position of the scroller. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollToMin	Scrolls to the minimum position of the scroller. (Inherited from IOracleFormsScrollable)
VerticalScrollbarStep	Increments the current scroll position by specified steps. (Inherited from IOracleFormsScrollable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsMenu Class

Description

Represents a top-level menu.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsMenu
 - [OracleFormsPopupMenu](#)

Syntax

```
'Declaration
Public Class OracleFormsMenu _
Inherits OracleFormsContainer _
Implements IOracleFormsMenuBase
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Checked	Whether the MenuItem is checked. (Inherited from IOracleFormsMenuBase)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)

Name	Description
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Check	Checks a menu item
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsChecked	a value that indicates whether the MenuItem is checked.
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects a menu item
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks a menu item
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsMenuItem Class

Description

Represents a selectable item inside a Menu.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsMenuItem

Syntax

```
'Declaration
Public Class OracleFormsMenuItem _
Inherits OracleFormsContainer _
Implements IOracleFormsMenuBase
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)

Name	Description
Checked	Whether the MenuItem is checked. (Inherited from IOracleFormsMenuBase)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks a menu item
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects a menu item
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks a menu item
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional

Name	Description
WaitForObject	<p>timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

OracleFormsPopList Class

Description

Represents a PopList.

Inheritance Hierarchy

- [OracleFormsChoice](#)
 - OracleFormsPopList

Syntax

```
'Declaration
Public Class OracleFormsPopList _
Inherits OracleFormsChoice
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
DropDownOpen	whether the drop down is currently open.
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
ItemCount	the number of items in the choice. (Inherited from OracleFormsChoice)
Items	the list of choice items. (Inherited from OracleFormsChoice)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	the index of the selected item. -1 (4Test: 0) is returned, if no item is currently selected. (Inherited from OracleFormsChoice)
SelectedItem	the selected item. (Inherited from OracleFormsChoice)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetItemText	Returns the item at the specified index. (Inherited from OracleFormsChoice)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>Select</i>	Selects the specified item in the choice. (Inherited from <i>OracleFormsChoice</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsPopupMenu Class

Description

Represents a context menu.

Inheritance Hierarchy

- [OracleFormsMenu](#)
 - OracleFormsPopupMenu

Syntax

```
'Declaration  
Public Class OracleFormsPopupMenu _  
Inherits OracleFormsMenu
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Checked	Whether the MenuItem is checked. (Inherited from IOracleFormsMenuBase)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks a menu item (Inherited from OracleFormsMenu)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsChecked	a value that indicates whether the MenuItem is checked. (Inherited from OracleFormsMenu)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects a menu item (Inherited from OracleFormsMenu)
SetFocus	Gives focus to the control. (Inherited from IFocusable)

Name	Description
 SetProperty 	Sets the value of the specified property. (Inherited from TestObject)
 StartScreenRecording 	Starts screen recording (Inherited from TestObject)
 StopScreenRecording 	Stops screen recording (Inherited from TestObject)
 TextCapture 	Returns the text in this object's visible area. (Inherited from TestObject)
 TextClick 	Clicks in the center of the specified text. (Inherited from TestObject)
 TextExists 	Returns whether the specified text exists. (Inherited from TestObject)
 TextRectangle 	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
 TypeKeys 	Sends a set of keystrokes to the object. (Inherited from IKeyable)
 TypePasswordKeys 	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
 Uncheck 	Unchecks a menu item (Inherited from OracleFormsMenu)
 WaitForChildDisappearance 	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForDisappearance 	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
 WaitForObject 	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF . Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
 WaitForProperty 	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsRadioButton Class

Description

Represents a radio button.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsRadioButton

Syntax

```
'Declaration
Public Class OracleFormsRadioButton _
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Checked	Whether the radio button is checked.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)

Name	Description
<i>Foreground</i>	The foreground color of the component. (Inherited from AWTComponent)
<i>Height</i>	The current height of the component. (Inherited from AWTComponent)
<i>Name</i>	The name of the component. (Inherited from AWTComponent)
<i>NativeHandle</i>	The native window handle for the object. (Inherited from INativeWindow)
<i>Text</i>	The text of the control. (Inherited from TestObject)
<i>Value</i>	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
<i>Width</i>	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
<i>CaptureBitmap</i>	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
<i>Check</i>	Checks the radio button.
<i>Click</i>	Clicks on the object. (Inherited from IClickable)
<i>DoubleClick</i>	Double-clicks a mouse button on the object. (Inherited from IClickable)
<i>Exists</i>	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from TestObject)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from TestObject)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsScrollbar Class

Description

Represents a scroll bar.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsScrollbar
 - [OracleFormsHorizontalScrollbar](#)
 - [OracleFormsVerticalScrollbar](#)

Syntax

```
'Declaration
Public Class OracleFormsScrollbar _
Inherits OracleFormsContainer _
Implements IOracleFormsScroller
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)

Name	Description
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScroller)
Position	the current position of the scroll bar. (Inherited from IOracleFormsScroller)
Range	the range value of the scroll bar. (Inherited from IOracleFormsScroller)
StepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScroller)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
Page	Increments the current scroll position by specified number of pages. (Inherited from IOracleFormsScroller)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
ScrollTo	Scrolls to the specified position. (Inherited from IOracleFormsScroller)
ScrollToMax	Scrolls to the maximum position of the scroller. (Inherited from IOracleFormsScroller)
ScrollToMin	Scrolls to the minimum position of the scroller. (Inherited from IOracleFormsScroller)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
Step	Increments the current scroll position by specified steps. (Inherited from IOracleFormsScroller)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsStatusArea Class

Description

Represents the area containing status information.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsStatusArea

Syntax

```
'Declaration  
Public Class OracleFormsStatusArea _  
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsStatusBar Class

Description

Represents a status bar.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsStatusBar

Syntax

```
'Declaration  
Public Class OracleFormsStatusBar _  
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsStatusBarItem Class

Description

Represents an item within a status bar.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsStatusBarItem

Syntax

```
'Declaration  
Public Class OracleFormsStatusBarItem _  
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsStatusIndicator Class

Description

Represents a status indicator.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsStatusIndicator

Syntax

```
'Declaration  
Public Class OracleFormsStatusIndicator _  
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsTabBar Class

Description

Represents a tab bar.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsTabBar

Syntax

```
'Declaration  
Public Class OracleFormsTabBar _  
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedTab	the currently selected tab.
TabCount	the number of tabs.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetTabIndex	Returns the tab index.
GetTabTitle	Returns the tab title.
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
IsTabEnabled	Returns whether or not the tab is currently enabled.
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the specified tab.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsTabBarItem Class

Description

Represents a tab bar item.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsTabBarItem

Syntax

```
'Declaration
Public Class OracleFormsTabBarItem _
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)

Name	Description
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)

Name	Description
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)

Name	Description
Select	Selects the tab.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An

Name	Description
WaitForPropertyNotEquals	exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsTabPanel Class

Description

Represents a container class for components.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsTabPanel

Syntax

```
'Declaration
Public Class OracleFormsTabPanel _
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)

Name	Description
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)

Name	Description
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider

Name	Description
WaitForObject	increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsTextField Class

Description

Represents an Oracle Forms text field.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsTextField

Syntax

```
'Declaration
Public Class OracleFormsTextField _
Inherits OracleFormsContainer _
Implements IOracleFormsScrollable
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)

Name	Description
CaretPosition	the position of the text insertion caret for the text field.
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Editable	whether the text of the text field can be changed by the user.
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
HorizontalScrollbarPageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarPosition	the current position of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarRange	the range value of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarStepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
IsPassword	whether the text field is a password text field.
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PasswordChar	the character that is displayed instead of the characters that are actually typed, if the text field is a password text field.
SelectedText	the current selected text.
SelectionEnd	the end position of the selected text.
SelectionStart	the start position of the selected text.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
VerticalScrollbarPageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarPosition	the current position of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarRange	the range value of the scroll bar. (Inherited from IOracleFormsScrollable)

Name	Description
VerticalScrollbarStepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Clear	Removes all the text from the text field.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalScrollbar	Gets the horizontal scroll bar. (Inherited from IOracleFormsScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalScrollbar	Gets the vertical scroll bar. (Inherited from IOracleFormsScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
HorizontalScrollbarPage	Increments the current scroll position by specified number of pages. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollTo	Scrolls to the specified position. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollToMax	Scrolls to the maximum position of the scroller. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollToMin	Scrolls to the minimum position of the scroller. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarStep	Increments the current scroll position by specified steps. (Inherited from IOracleFormsScrollable)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SelectAll	Selects all the text in the text field.
SetCaretPosition	Sets the position of the text insertion caret for this text field.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetSelectionRange	Selects the text within the specified range.
SetText	Sets the text of the text field to the specified text.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
VerticalScrollbarPage	Increments the current scroll position by specified number of pages. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollTo	Scrolls to the specified position. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollToMax	Scrolls to the maximum position of the scroller. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollToMin	Scrolls to the minimum position of the scroller. (Inherited from IOracleFormsScrollable)

Name	Description
VerticalScrollbarStep	Increments the current scroll position by specified steps. (Inherited from IOracleFormsScrollable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsTitleBar Class

Description

Represents a title bar.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsTitleBar

Syntax

```
'Declaration  
Public Class OracleFormsTitleBar _  
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsToolBar Class

Description

Represents a toolbar.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsToolBar

Syntax

```
'Declaration  
Public Class OracleFormsToolBar _  
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsToolBarItem Class

Description

Represents an element in a toolbar.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsToolBarItem

Syntax

```
'Declaration  
Public Class OracleFormsToolBarItem _  
Inherits OracleFormsContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
Select	Selects the element.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsTree Class

Description

Represents a tree.

Inheritance Hierarchy

- [OracleFormsContainer](#)
 - OracleFormsTree

Syntax

```
'Declaration  
Public Class OracleFormsTree _  
Inherits OracleFormsContainer _  
Implements IOracleFormsScrollable
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
HorizontalScrollbarPageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarPosition	the current position of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarRange	the range value of the scroll bar. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarStepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScrollable)

Name	Description
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedItemPath	the list of selected item path.
SelectedItemText	the selected item text.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
VerticalScrollbarPageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarPosition	the current position of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarRange	the range value of the scroll bar. (Inherited from IOracleFormsScrollable)
VerticalScrollbarStepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScrollable)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
ClickItem	Clicks on a tree node specified by the item path.
Collapse	Collapses the tree node identified by the specified item path.
CollapseAll	Collapses all tree nodes.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClickItem	Double-clicks on a tree node specified by the item path.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Expand	Expands the tree node identified by the specified item path.
ExpandAll	Expands all tree nodes.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalScrollbar	Gets the horizontal scroll bar. (Inherited from IOracleFormsScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalScrollbar	Gets the vertical scroll bar. (Inherited from IOracleFormsScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
HorizontalScrollbarPage	Increments the current scroll position by specified number of pages. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollTo	Scrolls to the specified position. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollToMax	Scrolls to the maximum position of the scroller. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarScrollToMin	Scrolls to the minimum position of the scroller. (Inherited from IOracleFormsScrollable)
HorizontalScrollbarStep	Increments the current scroll position by specified steps. (Inherited from IOracleFormsScrollable)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)

Name	Description
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsExpanded	Returns 'true' if the specified item path is expanded.
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
IsVisible	Returns 'true' if the specified item path is visible.
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from AWTComponent)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from AWTComponent)
ScrollItemIntoView	Scrolls the given item path to a visible position.
Select	Selects a tree node identified by the specified item path.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)

Name	Description
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
VerticalScrollbarPage	Increments the current scroll position by specified number of pages. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollTo	Scrolls to the specified position. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollToMax	Scrolls to the maximum position of the scroller. (Inherited from IOracleFormsScrollable)
VerticalScrollbarScrollToMin	Scrolls to the minimum position of the scroller. (Inherited from IOracleFormsScrollable)
VerticalScrollbarStep	Increments the current scroll position by specified steps. (Inherited from IOracleFormsScrollable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

OracleFormsVerticalScrollbar Class

Description

Represents a vertical scroll bar.

Inheritance Hierarchy

- [OracleFormsScrollbar](#)
 - OracleFormsVerticalScrollbar

Syntax

```
'Declaration
Public Class OracleFormsVerticalScrollbar _
Inherits OracleFormsScrollbar
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)

Name	Description
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
PageSize	the block increment value of the scroll bar. (Inherited from IOracleFormsScroller)
Position	the current position of the scroll bar. (Inherited from IOracleFormsScroller)
Range	the range value of the scroll bar. (Inherited from IOracleFormsScroller)
StepSize	the unit increment value of the scroll bar. (Inherited from IOracleFormsScroller)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>Page</i>	Increments the current scroll position by specified number of pages. (Inherited from <i>IOracleFormsScroller</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>ScrollTo</i>	Scrolls to the specified position. (Inherited from <i>IOracleFormsScroller</i>)
<i>ScrollToMax</i>	Scrolls to the maximum position of the scroller. (Inherited from <i>IOracleFormsScroller</i>)
<i>ScrollToMin</i>	Scrolls to the minimum position of the scroller. (Inherited from <i>IOracleFormsScroller</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>Step</i>	Increments the current scroll position by specified steps. (Inherited from <i>IOracleFormsScroller</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

SplitPaneDivider Class

Description

Represents a split pane divider.

Inheritance Hierarchy

- [AWTContainer](#)
 - SplitPaneDivider

Syntax

```
'Declaration  
Public Class SplitPaneDivider _  
Inherits AWTContainer
```

Properties

Name	Description
AccessibleName	The accessible name of the component. (Inherited from AWTComponent)
AccessibleRole	The accessible role of the component. (Inherited from AWTComponent)
Background	The background color of the component. (Inherited from AWTComponent)
Cursor	The name of the cursor set in the component. (Inherited from AWTComponent)
Enabled	whether the component is enabled. (Inherited from AWTComponent)
Focusable	whether the component can be focused. (Inherited from AWTComponent)
Font	The font of the component. (Inherited from AWTComponent)
Foreground	The foreground color of the component. (Inherited from AWTComponent)
Height	The current height of the component. (Inherited from AWTComponent)
Name	The name of the component. (Inherited from AWTComponent)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Width	The current width of the component. (Inherited from AWTComponent)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>AWTComponent</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the component into view if it is hidden and a child of a scroll pane. (Inherited from <i>AWTComponent</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Win32 Class Reference

Lists the available classes for testing Windows API-based controls.

AccessibleControl Class

Description

The class controls identified by the Accessibility extension.

Inheritance Hierarchy

- [BaseGuiTestObject](#)
 - AccessibleControl

Syntax

```
'Declaration  
Public Class AccessibleControl _  
Inherits BaseGuiTestObject
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Description	The description text of the accessible object.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Role	The role of this accessible object.
State	The state string that describes the different states of this accessible object.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoDefaultAction	Executes the default action of the underlying accessible control supports a default action.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)

Name	Description
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Header Class

Description

The class for list header controls. Headers are most commonly used in controls of the `ListViewW32` class.

Inheritance Hierarchy

- [Control](#)
 - Header

Syntax

```
'Declaration
Public Class Header _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
ColumnCount	The number of columns in the header.
ColumnOrder	The header's item indices as they are currently ordered.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <code>true</code> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DividerDoubleClick	Double-clicks the divider on the right side of the specified item.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleClick	Double-clicks an item.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetColumnTitle	Returns the title of a specified header item.
GetColumnWidth	Returns the width of a specified column.
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Resize	Resizes an item of the header.
Select	Selects an item or button.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForScreenshotStable	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

Listview Class

Description

The class for elaborated list controls that support multiple ways of viewing a list of nonhierarchical data. These views include large and small rearrangable icons and a detail (or report) view.

Inheritance Hierarchy

- [Control](#)
 - ListView

Syntax

```
'Declaration
Public Class ListView _
Inherits Control
```

Properties

Name	Description
AllowsCheck	Whether this list view supports checking items.
AllowsMultiSelect	Whether the list view supports selecting multiple items.
Background	The background color of the GUI object. (Inherited from GuiTestObject)
CheckedIndices	A list containing the indices of the checked items.
CheckedItems	A list containing the checked items.
ColumnCount	The number of columns per item.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items contained in the list view.
Items	A list containing all items of the listview.

Name	Description
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndices	A list containing the indices of the selected items.
SelectedItems	A list containing the selected items.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Check	Checks an item.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleSelect	Double-clicks an item.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
ExtendSelect	Selects a range of items by extending the selection in the extend-selection listview.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)

Name	Description
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetContents	Returns the contents of a listview control. Each returned item is delimited by a semicolon. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetItemImageIndex	Returns the index into the image list of a list view that contains icons. This allows to verify that the correct image is used for an item.
GetItemIndex	Returns the index of the list view item or -1 if is not found.
GetItemRect	Returns rectangle of the item. If <code>getIconRect</code> is set then the rectangle of the icon of the item will be returned otherwise the rectangle of the item text is returned. If a <code>columnIndex</code> is specified the rectangle of this column is returned.
GetItemText	Returns the text of an item.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MultiSelect</i>	Adds an item in the list to the set of currently selected items.
<i>MultiUnselect</i>	Removes an item in the list from the set of selected items.
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Select</i>	Selects an item.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetItemFocus</i>	Gives focus to the item in the list.
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Unchecks an item.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

MonthCalendar Class

Description

The class that selects the calendar date and time.

Inheritance Hierarchy

- [Control](#)
 - [MonthCalendar](#)

Syntax

```
'Declaration  
Public Class MonthCalendar _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
ExtendSelect	Selects a range of dates and times.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
OpenMonthMenu	Opens a calendar menu to a specific month.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects a date and time.
SelectNextMonth	Selects the next month.
SelectPreviousMonth	Selects the previous month.
SelectToday	Selects the date for today.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Pager Class

Description

The class for tabbed, multi-page dialogs and for button bars. On Windows, this class corresponds to the TabControl control. On Windows, only major tabs are supported. Minor tabs (also called 'subpages' or 'child pages') are not supported.

Inheritance Hierarchy

- [Control](#)
 - Pager

Syntax

```
'Declaration  
Public Class Pager _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
Horizontal	Whether the up/down control is horizontal.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is

Name	Description
	<p>%LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollDown	Moves the scroll bar down.
ScrollUp	Moves the scroll bar up.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ProgressBar Class

Description

The class for progress bar controls. A progress bar displays a progress.

Inheritance Hierarchy

- [Control](#)
 - ProgressBar

Syntax

```
'Declaration  
Public Class ProgressBar _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Position	The current position.
Range	The range of the progress bar control.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The

Name	Description
	<p>captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)</p>
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

StatusBar Class

Description

The class for status bar controls. A status bar is a container for a set of text labels that change dynamically.

Inheritance Hierarchy

- [Control](#)
 - StatusBar

Syntax

```
'Declaration  
Public Class StatusBar _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel

Name	Description
	must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <code>timeout</code> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <code>false</code> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)

Name	Description
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

UpDown Class

Description

The class for up/down controls.

Inheritance Hierarchy

- [Control](#)
 - UpDown
 - [NumericUpDown](#)

Syntax

```
'Declaration  
Public Class UpDown _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
Horizontal	Whether the up/down control is horizontal.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Position	The position for the up/down control.
Range	The range of the up/down control.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
WrapAround	Whether values are wrapped if smaller than minimum or larger than maximum.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when

Name	Description
	parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Decrement	Decreases the position value of the up/down control.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject . (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Increment</i>	Increases the position value of the up/down control.
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>SetFocus</i>	Gives focus to the control. (Inherited from IFocusable)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from TestObject)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from TestObject)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from TestObject)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from TestObject)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from TestObject)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from TestObject)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Windows Forms Class Reference

Lists the available classes for testing Windows Forms controls.

CheckedListBox Class

Description

CheckedListBox represents list boxes that contain a check box.

Inheritance Hierarchy

- [ListBox](#)
 - CheckedListBox

Syntax

```
'Declaration  
Public Class CheckedListBox _  
Inherits ListBox
```

Properties

Name	Description
AllowsMultiSelect	Whether the control supports selecting multiple items. (Inherited from ListBox)
Background	The background color of the GUI object. (Inherited from GuiTestObject)
CheckedIndices	The indices of the checked item(s).
CheckedItems	The names of the checked item(s).
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items in the list box. (Inherited from ListBox)
Items	A list of items in the list box. (Inherited from ListBox)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	The index of the first selected item. (Inherited from ListBox)
SelectedIndices	The indices of the selected item(s). (Inherited from ListBox)
SelectedItem	The name of the first selected item. (Inherited from ListBox)
SelectedItems	The names of the selected item(s). (Inherited from ListBox)

Name	Description
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Selects an item from the checked list box.
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleSelect	Double-clicks an item in the list box. (Inherited from ListBox)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
ExtendSelect	Selects a range of items by extending the selection in the list box. (Inherited from ListBox)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)

Name	Description
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetHorizontalBar</i>	Returns the horizontal scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MultiSelect</i>	Selects an item in the multi- or extend-selection list box. (Inherited from <i>ListBox</i>)
<i>MultiUnselect</i>	Unselects an item in the multi- or extend-selection list box. (Inherited from <i>ListBox</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Select	Selects an item from the list box. (Inherited from ListBox)
SelectRange	Selects a range of items in the extend-selection list box. (Inherited from ListBox)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Uncheck	Selects an item from the checked list box.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException

Name	Description
	is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DataGrid Class

Description

The class for datagrids and data grid view.

Inheritance Hierarchy

- [Control](#)
 - DataGrid

Syntax

```
'Declaration
Public Class DataGrid _
Inherits Control
```

Properties

Name	Description
AllowsMultiSelect	Whether the table supports selecting multiple items.

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
ColumnCount	The number of columns in the DataGrid.
ColumnItems	A list of all items in the column.
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
Items	A list of all items in the column.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
RowCount	The number of rows in the DataGrid.
RowItems	A list of all items in the row.
SelectedItems	The selected item(s).
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
DoubleSelectedItem	Double-clicks an item.

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
ExtendSelectedItem	Expands the selection to the given item.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)

Name	Description
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
MultiSelectItem	Adds a row from the table to the set of selected rows.
MultiUnselectItem	Removes a item from the set of selected items.
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Tries to bring the item to the top left corner of the datagrid.
SelectAllItems	Selects all items in the datagrid.
SelectItem	Selects an item.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DataGridColumn Class

Description

The class for columns in a table.

Inheritance Hierarchy

- [Item](#)
 - DataGridViewColumn

Syntax

```
'Declaration  
Public Class DataGridViewColumn _  
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
Width	The width of the column.

Methods

Name	Description
AutoSize	Automatically resizes a column by double clicking its divider.
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)

Name	Description
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Resize</i>	Resizes a column.
<i>ScrollIntoView</i>	Tries to bring the column to the left by scrolling the data grid.
<i>Select</i>	Selects a column.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DataGridItem Class

Description

The class for datagrids and data grid view rows.

Inheritance Hierarchy

- [Item](#)
 - DataGridItem

Syntax

```
'Declaration  
Public Class DataGridViewItem _  
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Column	The column of the item in the data grid
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Row	The row of the item in the data grid
Selected	Whether the item is selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Tries to bring the item to the top left corner of the datagrid.
<i>Select</i>	Selects an item.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
WaitForObject	<p>OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p> <p>Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code>. Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code>, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)</p>
WaitForProperty	<p>Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForPropertyNotEquals	<p>Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>
WaitForScreenshotStable	<p>Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)</p>

DataGridRow Class

Description

The class for rows in a table.

Inheritance Hierarchy

- [Item](#)
 - DataGridRow

Syntax

```
'Declaration
Public Class DataGridRow _
Inherits Item
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
Height	The width of the column.
ItemCount	The number of items in the row.
Items	A list of items in the row.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Resize	Resizes a column.
ScrollIntoView	Tries to bring the row to the top by scrolling the data grid.
Select	Selects a column.
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change

Name	Description
	the timeout by setting the value of the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> . Use <i>WaitForObject</i> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <i>WaitForObject</i> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <i>WaitForObject</i> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_RESOLVE_OBJDEF</i> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <i>OPT_WAIT_SCREENSHOT_STABLE</i> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomainUpDown Class

Description

The class for up/down controls.

Inheritance Hierarchy

- [Control](#)
 - DomainUpDown

Syntax

```
'Declaration
Public Class DomainUpDown _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)

Name	Description
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
ItemCount	The number of items contained in the list view.
Items	The items in the domain up/down control.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
SelectedIndex	The position for the up/down control.
SelectedItem	The position for the up/down control.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
WrapAround	Whether values are wrapped if smaller than minimum or larger than maximum.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Decrement	Decreases the position value of the up/down control.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Increment	Increases the position value of the up/down control.
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>Select</i>	Selects an item.
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>IFocusable</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from <i>TestObject</i>)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from <i>IKeyable</i>)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from <i>IKeyable</i>)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from <i>TestObject</i>)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

ElementHost Class

Description

A Windows Forms control that can be used to host a Windows Presentation Foundation (WPF) element.

Inheritance Hierarchy

- [Control](#)
 - ElementHost

Syntax

```
'Declaration  
Public Class ElementHost _  
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetHorizontalBar</i>	Returns the horizontal scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)

Name	Description
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
<i>SetFocus</i>	Gives focus to the control. (Inherited from IFocusable)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from TestObject)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from TestObject)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from TestObject)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from TestObject)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from TestObject)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from TestObject)
<i>TextRectangle</i>	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
<i>TypeKeys</i>	Sends a set of keystrokes to the object. (Inherited from IKeyable)
<i>TypePasswordKeys</i>	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
<i>WaitForChildDisappearance</i>	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
<i>WaitForDisappearance</i>	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForObject	Waits for an object that matches the specified locator. If no object matches within a timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FormsHost Class

Description

An element that allows you to host a Windows Forms control on a WPF page.

Inheritance Hierarchy

- [Control](#)
 - FormsHost

Syntax

```
'Declaration
Public Class FormsHost _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)

Name	Description
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from TestObject)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from TestObject)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from TestObject)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
<i>GetHorizontalBar</i>	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from TestObject)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from TestObject)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from IScrollable)
<i>HighlightObject</i>	Highlights this object. (Inherited from TestObject)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from TestObject)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from TestObject)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from TestObject)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from TestObject)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from TestObject)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from IFocusable)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT

Name	Description
	takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

FormsWindow Class

Description

FormsWindow is the class that represents the top-level window of a Windows Forms application.

Inheritance Hierarchy

- [Window](#)
 - FormsWindow

Syntax

```
'Declaration
Public Class FormsWindow _
Inherits Window
```

Properties

Name	Description
Application	The name of the Application that this Window belongs to. (Inherited from Window)

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)

Name	Description
<i>Exists</i>	Checks if an object exists in the application under test. (Inherited from <i>TestObject</i>)
<i>Find</i>	Finds an object specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>FindAll</i>	Finds all objects specified by an XPath locator. (Inherited from <i>TestObject</i>)
<i>GenerateLocator</i>	Returns a locator for this object. (Inherited from <i>TestObject</i>)
<i>GetChildren</i>	Returns the child objects of this object. (Inherited from <i>TestObject</i>)
<i>GetCloseWindows</i>	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from <i>IMoveable</i>)
<i>GetDynamicMethodList</i>	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from <i>TestObject</i>)
<i>GetFocus</i>	Returns the object with the input focus. (Inherited from <i>IMoveable</i>)
<i>GetHorizontalBar</i>	Returns the horizontal scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>GetNextCloseWindow</i>	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from <i>IMoveable</i>)
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsActive	Returns whether the window is set to active. (Inherited from IMoveable)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
LoadAssembly	Loads the assembly on the specified path in the Windows Forms application.
Maximize	Maximizes the window. (Inherited from IMoveable)
Minimize	Reduces the window to an icon. (Inherited from IMoveable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)

Name	Description
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional

Name	Description
	timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

MenuStrip Class

Description

MenuStrip is the class that creates a new menu strip for a Windows Forms application.

Inheritance Hierarchy

- [Control](#)
 - MenuStrip

Syntax

```
'Declaration
Public Class MenuStrip _
Inherits Control
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetVerticalBar	Returns the vertical scroll bar for this control. (Inherited from IScrollable)

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsFocused	Return whether the control has focus. (Inherited from IFocusable)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default

Name	Description
	interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

NumericUpDown Class

Description

The class for numeric up/down controls.

Inheritance Hierarchy

- [UpDown](#)
 - NumericUpDown

Syntax

```
'Declaration
Public Class NumericUpDown _
Inherits UpDown
```

Properties

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
Horizontal	Whether the up/down control is horizontal. (Inherited from UpDown)
Maximum	The maximum value of the up/down control.
Minimum	The minimum value of the up/down control.
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Position	The position for the up/down control. (Inherited from UpDown)
PositionDecimal	The position for the up/down control.
Range	The range of the up/down control. (Inherited from UpDown)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need

Name	Description
WrapAround	<p>this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)</p> <p>Whether values are wrapped if smaller than minimum or larger than maximum. (Inherited from UpDown)</p>

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
Decrement	Decreases the position value of the up/down control. (Inherited from UpDown)
DecrementDecimal	Decreases the position value of the up/down control.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)

Name	Description
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Increment</i>	Increases the position value of the up/down control. (Inherited from <i>UpDown</i>)
<i>IncrementDecimal</i>	Increases the position value of the up/down control.
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>OpenContextMenu</i>	Opens a context menu at the specified position. (Inherited from <i>BaseGuiTestObject</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)

Name	Description
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

xBrowser Class Reference

Lists the available classes for testing xBrowser controls.

BrowserApplication Class

Description

Represents the top-level window of a browser application.

Note: BrowserApplication is derived from BaseGuiTestObject and inherits the two scrollbar methods GetHorizontalBar and GetVerticalBar. However, these two methods do not work on a BrowserApplication.

Inheritance Hierarchy

- [Window](#)
 - BrowserApplication

Syntax

```
'Declaration
Public Class BrowserApplication _
Inherits Window
```

Properties

Name	Description
Application	The name of the Application that this Window belongs to. (Inherited from Window)

Name	Description
Background	The background color of the GUI object. (Inherited from GuiTestObject)
Enabled	Whether the GUI object is enabled. (Inherited from GuiTestObject)
Font	The font type of the GUI object. (Inherited from GuiTestObject)
Foreground	The foreground color of the GUI object. (Inherited from GuiTestObject)
NativeHandle	The native window handle for the object. (Inherited from INativeWindow)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from GuiTestObject)
WindowState	The state of a window. Values include: 1=minimized, 2=maximized, 3=restored (Inherited from IMoveable)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ClearCache	Clears the browser's cache.
Click	Clicks on the object. (Inherited from IClickable)
Close	Closes the window. (Inherited from IMoveable)
CloseOtherTabs	Closes all tabs except the active one. When several tabs are open, Internet Explorer only displays some of the tabs and scroll buttons. This method closes all other tabs except for the selected one, including the tabs that are not currently visible. This method is not supported for mobile web applications.
CloseSynchron	Closes the window and waits until the window is closed. (Inherited from IMoveable)

Name	Description
CloseTab	Closes the specified tab. If no tab is specified, the active tab is closed. When several tabs are open, Internet Explorer only displays some of the tabs and scroll buttons. As a result, only the currently visible tabs work with this method, which affects index semantics. For example, only the currently visible tabs can be closed. Index 0 identifies the first visible tab. This method is not supported for mobile Web applications.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
ExistsTab	Returns true if the tab exists and false otherwise. This method is not supported for mobile web applications.
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetActiveTab	Returns the active tab. This method is not supported for mobile Web applications.
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCloseWindows	Returns the windows that need to be closed in order so that only the main window of the application is open. (Inherited from IMoveable)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetFocus	Returns the object with the input focus. (Inherited from IMoveable)
GetHorizontalBar	Returns the horizontal scroll bar for this control. (Inherited from IScrollable)
GetNextCloseWindow	Returns the next window that need to be closed in order to close all windows of the application except the main window. (Inherited from IMoveable)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)

Name	Description
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetSelectedTab</i>	Returns the currently selected tab. This method is not supported for mobile Web applications.
<i>GetSelectedTabIndex</i>	Returns the index of the active tab. When several tabs are open, Internet Explorer only displays some of the tabs and scroll buttons. As a result, this method only returns the index relative to the currently visible tabs. Index 0 identifies the first visible tab. This method is not supported for mobile web applications.
<i>GetSelectedTabName</i>	Returns the name of the active tab. This method is not supported for mobile web applications.
<i>GetTabCount</i>	Returns the number of open tabs. When several tabs are open, Internet Explorer only displays some of the tabs and scroll buttons. As a result, this method only returns the number of currently visible tabs. This method is not supported for mobile web applications.
<i>GetVerticalBar</i>	Returns the vertical scroll bar for this control. (Inherited from <i>IScrollable</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>IsActive</i>	Returns whether the window is set to active. (Inherited from <i>IMoveable</i>)
<i>IsFocused</i>	Return whether the control has focus. (Inherited from <i>IFocusable</i>)
<i>Maximize</i>	Maximizes the window. (Inherited from <i>IMoveable</i>)
<i>Minimize</i>	Reduces the window to an icon. (Inherited from <i>IMoveable</i>)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Move	Moves the window. (Inherited from IMoveable)
Navigate	Navigate to the specified URL in the currently active tab/window
OpenContextMenu	Opens a context menu at the specified position. (Inherited from BaseGuiTestObject)
OpenTab	Opens a new tab and navigates to the given URL. If no URL is given, the tab navigates to about:blank. This method is not supported for mobile Web applications.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Restore	Restores the window to its previous size. (Inherited from IMoveable)
SelectTab	Selects the specified tab. When several tabs are open, Internet Explorer only displays some of the tabs and scroll buttons. As a result, only the currently visible tabs work with this method, which affects index semantics. For example, only the currently visible tabs can be selected. Index 0 selects the first visible tab. This method is not supported for mobile Web applications.
SetActive	Makes the window active. (Inherited from IMoveable)
SetFocus	Gives focus to the control. (Inherited from IFocusable)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
Size	Resizes the window. (Inherited from IMoveable)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)

Name	Description
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

BrowserObject Class

Description

Base class for all objects that are contained within a BrowserApplication.

Inheritance Hierarchy

- [TestObject](#)
 - BrowserObject
 - [BrowserWindow](#)
 - [DomElement](#)

Syntax

```
'Declaration
Public Class BrowserObject _
Inherits TestObject _
Implements IClickable, IKeyable
```

Properties

Name	Description
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay.

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires.

Name	Description
	If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)

Name	Description
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
SetFocus	Gives focus to the control.
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> ,

Name	Description
WaitForObject	because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

BrowserWindow Class

Description

A browser window is the top-level container for a web page. It exposes the DOM tree through DomElements. E.g. a tab in IE7, an embedded browser control in SWT.

Inheritance Hierarchy

- [BrowserObject](#)
 - BrowserWindow

Syntax

```
'Declaration
Public Class BrowserWindow _
Inherits BrowserObject
```

Properties

Name	Description
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
ViewportHeight	The height of the browser viewport

Name	Description
ViewportName	The name of the viewport that matches the innerWidth and innerHeight of the browser window
ViewportOrientation	The orientation of the browser viewport, this can be portrait or landscape
ViewportWidth	The width of the browser viewport
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
AcceptAlert	Accepts a JavaScript alert by pressing OK.
Back	Goes back in the history.
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
CaptureFullPageBitmap	Capture a screenshot of the whole web page and save it as a file to the specified path. The filename extension determines the file type. Possible extensions are .jpg, .jpeg, .png, and .bmp.
Click	Clicks on the object. (Inherited from IClickable)
Close	Close a modal browser dialog, a tab of the browser, or the browser itself if it is the last tab (or IE6).
DismissAlert	Dismisses a JavaScript alert.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates JavaScript code within the top-level document context.
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
Forward	Goes forward in the history.
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetAlertText	Retrieves the message text of a JavaScript alert.
GetBrowserMajorVersion	Returns the browser's major version number.
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of the BrowserWindow in HTML pixels. The click function also uses HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not affected by changing zoom levels on mobile browsers.
GetPageSource	Returns the page source of the web site
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetUrl	Returns the URL for the main document.
GetUserAgent	Returns the browser's user agent string
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)

Name	Description
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsAlertPresent	Returns true if a JavaScript alert is open
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
Navigate	Navigates the main document to an URL.
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
Reload	Reloads the main document as the reload button in the browser does.
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetViewportName	Set the size of the browser window to the specified size from your Browser Size list.
SetViewportSize	Sets the size of the browser window to the specified width and height.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomButton Class

Description

A DomButton represents all DOM elements that were specified using one of the following tags: <input type='submit'>, <input type='reset'>, <input type='button'> or <button>.

Inheritance Hierarchy

- [DomElement](#)
 - DomButton

Syntax

```
'Declaration  
Public Class DomButton _  
Inherits DomElement
```

Properties

Name	Description
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)

Name	Description
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetText	Gets the visible text of a DOM element. (Inherited from DomElement)
Highlight	Highlights a DOM element. (Inherited from DomElement)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the DOM element into the visible area of the browser window. (Inherited from DomElement)
Select	Clicks a button.
SetDomAttribute	Sets the value of an object specific DOM attribute. (Inherited from DomElement)
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)

Name	Description
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't

Name	Description
WaitForScreenshotStable	explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomCheckBox Class

Description

A DomCheckBox represents all DOM elements that were specified using <input type='checkbox'> tag.

Inheritance Hierarchy

- [DomElement](#)
 - DomCheckBox

Syntax

```
'Declaration
Public Class DomCheckBox _
Inherits DomElement
```

Properties

Name	Description
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
State	The state of a check box. Values include: 1=checked, 2=unchecked,3=undecided
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Check	Checks the check box.
Click	Clicks on the object. (Inherited from IClickable)
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)

Name	Description
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetText	Gets the visible text of a DOM element. (Inherited from DomElement)
Highlight	Highlights a DOM element. (Inherited from DomElement)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)

Name	Description
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
IsChecked	Returns <i>true</i> if the checkbox is checked and <i>false</i> otherwise.
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the DOM element into the visible area of the browser window. (Inherited from DomElement)
Select	Selects the check box
SetDomAttribute	Sets the value of an object specific DOM attribute. (Inherited from DomElement)
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetUndecided	Sets the check box to the undecided state.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
Toggle	Toggles the state of the check box. If the checkbox is checked it is unchecked and vice versa. If the checkbox is in the undecided state it is left undecided.
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
Uncheck	Unchecks the check box.
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomDataList Class

Description

A DomDataList represents an input element with pre-defined values in a <datalist> element.

Inheritance Hierarchy

- [DomTextField](#)
 - DomDataList

Syntax

```
'Declaration  
Public Class DomDataList _  
Inherits DomTextField _  
Implements IItemSelectable
```

Properties

Name	Description
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
IsPassword	whether the text field is a password text field. (Inherited from DomTextField)
ItemCount	the number of items in the listbox. (Inherited from IItemSelectable)
Items	the list of items in the listbox. (Inherited from IItemSelectable)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
ClearText	Clears the textfield's text (Inherited from DomTextField)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses

Name	Description
	HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetText	Gets the visible text of a DOM element. (Inherited from DomElement)
Highlight	Highlights a DOM element. (Inherited from DomElement)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the DOM element into the visible area of the browser window. (Inherited from DomElement)
Select	Selects an item. (Inherited from IItemSelectable)
SetDomAttribute	Sets the value of an object specific DOM attribute. (Inherited from DomElement)
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetText	Sets the text. (Inherited from DomTextField)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when

Name	Description
	a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomElement Class

Description

A `DomElement` represents any element in the DOM tree.

Inheritance Hierarchy

- [BrowserObject](#)
 - `DomElement`
 - [DomButton](#)
 - [DomCheckBox](#)
 - [DomEmbeddedElement](#)
 - [DomForm](#)
 - [DomLink](#)
 - [DomListBox](#)
 - [DomRadioButton](#)
 - [DomTable](#)
 - [DomTableRow](#)
 - [DomTextField](#)

Syntax

```
'Declaration  
Public Class DomElement _  
Inherits BrowserObject
```

Properties

Name	Description
IsFocused	Whether the DOM element has focus.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses.
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications.
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications.
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context.

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name.
GetDomAttribute	Gets the value of an object specific DOM attribute.
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications.
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers.
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetText	Gets the visible text of a DOM element.
Highlight	Highlights a DOM element.

Name	Description
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the DOM element into the visible area of the browser window.
SetDomAttribute	Sets the value of an object specific DOM attribute.
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)

Name	Description
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Unhighlight	Restores the original foreground and background colors.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
	OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomEmbeddedElement Class

Description

A DOM element that was specified by an <embed> or <object> tag.

Inheritance Hierarchy

- [DomElement](#)
 - DomEmbeddedElement

Syntax

```
'Declaration
Public Class DomEmbeddedElement _
Inherits DomElement
```

Properties

Name	Description
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)

Name	Description
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses

Name	Description
	HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetText	Gets the visible text of a DOM element. (Inherited from DomElement)
Highlight	Highlights a DOM element. (Inherited from DomElement)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)

Name	Description
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the DOM element into the visible area of the browser window. (Inherited from DomElement)
SetDomAttribute	Sets the value of an object specific DOM attribute. (Inherited from DomElement)
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random

Name	Description
WaitForObject	timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomForm Class

Description

A DomForm represents all DOM elements that were specified using the <form> tag. All methods and properties in this class are not supported for mobile web applications.

Inheritance Hierarchy

- [DomElement](#)
 - DomForm

Syntax

```
'Declaration
Public Class DomForm _
Inherits DomElement
```

Properties

Name	Description
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)

Name	Description
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)

Name	Description
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetText	Gets the visible text of a DOM element. (Inherited from DomElement)
Highlight	Highlights a DOM element. (Inherited from DomElement)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the DOM element into the visible area of the browser window. (Inherited from DomElement)
SetDomAttribute	Sets the value of an object specific DOM attribute. (Inherited from DomElement)
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
Submit	Submits the form.
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)

Name	Description
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomLink Class

Description

A DomLink represents all DOM elements that were specified using the <a> tag.

Inheritance Hierarchy

- [DomElement](#)
 - DomLink

Syntax

```
'Declaration  
Public Class DomLink _  
Inherits DomElement
```

Properties

Name	Description
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay

Name	Description
	for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)

Name	Description
<i>GetParent</i>	Looks up the parent of this object in the test application. (Inherited from <i>TestObject</i>)
<i>GetProperty</i>	Returns the value of the specified property. (Inherited from <i>TestObject</i>)
<i>GetPropertyList</i>	Returns a list of property names that can be retrieved for the given object. (Inherited from <i>TestObject</i>)
<i>GetRect</i>	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from <i>TestObject</i>)
<i>GetText</i>	Gets the visible text of a DOM element. (Inherited from <i>DomElement</i>)
<i>Highlight</i>	Highlights a DOM element. (Inherited from <i>DomElement</i>)
<i>HighlightObject</i>	Highlights this object. (Inherited from <i>TestObject</i>)
<i>ImageClick</i>	Clicks on specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageClickFile</i>	Clicks on the specified image. (Inherited from <i>TestObject</i>)
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the DOM element into the visible area of the browser window. (Inherited from <i>DomElement</i>)

Name	Description
Select	Clicks a link.
SetDomAttribute	Sets the value of an object specific DOM attribute. (Inherited from DomElement)
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomListBox Class

Description

A DomListBox represents all DOM elements that were specified using the <select> tag.

Inheritance Hierarchy

- [DomElement](#)
 - DomListBox

Syntax

```
'Declaration
Public Class DomListBox _
Inherits DomElement _
Implements IItemSelectable
```

Properties

Name	Description
AllowsMultiSelect	whether the control is multiselectable.
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
ItemCount	the number of items in the listbox. (Inherited from IItemSelectable)
Items	the list of items in the listbox. (Inherited from IItemSelectable)
SelectedIndices	The indices of the selected item(s).

Name	Description
SelectedItems	The names of the selected item(s).
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)

Name	Description
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetText	Gets the visible text of a DOM element. (Inherited from DomElement)
Highlight	Highlights a DOM element. (Inherited from DomElement)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)

Name	Description
<i>ImageExists</i>	Returns whether the specified image asset exists. (Inherited from <i>TestObject</i>)
<i>ImageExistsFile</i>	Returns whether the specified image exists. (Inherited from <i>TestObject</i>)
<i>ImageRectangle</i>	Returns the object-relative rectangle of the specified image asset. (Inherited from <i>TestObject</i>)
<i>ImageRectangleFile</i>	Returns the object-relative rectangle of the specified image. (Inherited from <i>TestObject</i>)
<i>Invoke</i>	Dynamically invokes a method on the test object. (Inherited from <i>TestObject</i>)
<i>InvokeMethods</i>	Dynamically invokes a sequence of methods starting at this <i>TestObject</i> . (Inherited from <i>TestObject</i>)
<i>MouseMove</i>	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>MultiSelect</i>	Selects a list of items.
<i>PressKeys</i>	Presses (but does not release) a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>PressMouse</i>	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ReleaseKeys</i>	Releases a set of keys or mouse buttons. (Inherited from <i>IKeyable</i>)
<i>ReleaseMouse</i>	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from <i>IClickable</i>)
<i>ScrollIntoView</i>	Scrolls the DOM element into the visible area of the browser window. (Inherited from <i>DomElement</i>)
<i>Select</i>	Selects an item. (Inherited from <i>IItemSelectable</i>)
<i>SetDomAttribute</i>	Sets the value of an object specific DOM attribute. (Inherited from <i>DomElement</i>)
<i>SetFocus</i>	Gives focus to the control. (Inherited from <i>BrowserObject</i>)
<i>SetProperty</i>	Sets the value of the specified property. (Inherited from <i>TestObject</i>)
<i>StartScreenRecording</i>	Starts screen recording (Inherited from <i>TestObject</i>)
<i>StopScreenRecording</i>	Stops screen recording (Inherited from <i>TestObject</i>)
<i>TextCapture</i>	Returns the text in this object's visible area. (Inherited from <i>TestObject</i>)
<i>TextClick</i>	Clicks in the center of the specified text. (Inherited from <i>TestObject</i>)
<i>TextExists</i>	Returns whether the specified text exists. (Inherited from <i>TestObject</i>)

Name	Description
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
Unselect	Unselects an item.
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option

Name	Description
	OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomRadioButton Class

Description

A DomRadioButton represents all DOM elements that were specified using <input type='radio'> tag.

Inheritance Hierarchy

- [DomElement](#)
 - DomRadioButton

Syntax

'Declaration

Public Class DomRadioButton _

Inherits DomElement

Properties

Name	Description
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
RadioListItemCount	The number of items in the radiolist. This property is not supported for mobile web applications.
RadioListItems	the list of items in the radio list
RadioListSelectedIndex	The index of the selected item. This property is not supported for mobile web applications.
RadioListSelectedItem	The item that is selected in the radiolist. This property is not supported for mobile web applications.
Selected	whether the radio button is selected.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)

Name	Description
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetText	Gets the visible text of a DOM element. (Inherited from DomElement)
Highlight	Highlights a DOM element. (Inherited from DomElement)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)

Name	Description
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
RadioListSelect	Selects an item in the radio list.
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the DOM element into the visible area of the browser window. (Inherited from DomElement)
Select	Selects the radio button.
SetDomAttribute	Sets the value of an object specific DOM attribute. (Inherited from DomElement)
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)

Name	Description
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_SCREENSHOT_STABLE</code> will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomTable Class

Description

A `DomTable` represents the `<table>` tag. All methods and properties in this class are not supported for mobile web applications.

Inheritance Hierarchy

- [DomElement](#)
 - `DomTable`

Syntax

```
'Declaration  
Public Class DomTable _  
Inherits DomElement
```

Properties

Name	Description
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <i>System.Drawing.Imaging.PixelFormat.Format32bppRgb</i> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)

Name	Description
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCell	Returns the specified cell.
GetCellText	Returns the content of a cell element.
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetColumnCount	Number of columns of the table.
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)

Name	Description
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetRow	Returns the DOM element for the specified row.
GetRowCount	Returns the number of rows
GetRows	Returns a list of rows (DomTableRows)
GetRowText	Returns the content of all cell elements of a specified row.
GetText	Gets the visible text of a DOM element. (Inherited from DomElement)
Highlight	Highlights a DOM element. (Inherited from DomElement)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the DOM element into the visible area of the browser window. (Inherited from DomElement)

Name	Description
SetDomAttribute	Sets the value of an object specific DOM attribute. (Inherited from DomElement)
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)

Name	Description
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached.If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomTableRow Class

Description

A DomTableRow represents the <tr> tag. All methods and properties in this class are not supported for mobile web applications.

Inheritance Hierarchy

- [DomElement](#)
 - DomTableRow

Syntax

```
'Declaration
Public Class DomTableRow _
Inherits DomElement
```

Properties

Name	Description
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application

Name	Description
	under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
Click	Clicks on the object. (Inherited from IClickable)
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use MouseMove function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetCell	Returns the DOM element in the given cell.

Name	Description
GetCellCount	Returns the number of cells in this table row
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this TestObject. (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a DomElement relative to the BrowserWindow in HTML pixels. In contrast to the screen coordinates that are provided by the GetRect function, the coordinates provided by GetHtmlRect are not affected by the zoom level on mobile browsers. The click function also uses HTML pixel coordinates for all DomElements. We recommend to use GetHtmlRect for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetText	Gets the visible text of a DOM element. (Inherited from DomElement)
Highlight	Highlights a DOM element. (Inherited from DomElement)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)

Name	Description
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the DOM element into the visible area of the browser window. (Inherited from DomElement)
SetDomAttribute	Sets the value of an object specific DOM attribute. (Inherited from DomElement)
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)

Name	Description
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an ObjectNotFoundException is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option OPT_WAIT_RESOLVE_OBJDEF. Use WaitForObject if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a WaitForObject, because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding WaitForObject statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForPropertyNotEquals	Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

DomTextField Class

Description

A DomTextField represents DOM elements that were specified using one of the following tags: <input type='text'>, <input type='password'>, <input type='file'> or <textarea>.

Inheritance Hierarchy

- [DomElement](#)
 - DomTextField
 - [DomDataList](#)

Syntax

```
'Declaration  
Public Class DomTextField _  
Inherits DomElement
```

Properties

Name	Description
IsFocused	Whether the DOM element has focus. (Inherited from DomElement)
IsPassword	whether the text field is a password text field.
Text	The text of the control. (Inherited from TestObject)
Value	The value of the control, e.g.: text in a text control. (Inherited from TestObject)
Visible	Whether the object is visible. You can only locate visible objects, so by default the value is always <i>true</i> . However, you might need this property when an already located object in the application under test gets invisible during replay. (Inherited from BrowserObject)

Methods

Name	Description
CaptureBitmap	Saves a bitmap image of this object to a file. If you do not specify an absolute file name, the bitmap is saved to the default bitmap location on the machine that runs the Agent, which is %LOCALAPPDATA%/Silk/SilkTest/capturedBitmaps. The captured bitmap's pixels are stored in 32bit RGB order; when parsing the image the 4th byte (alpha channel) of every pixel must be ignored as it is used only for padding. In .NET, the constant <code>System.Drawing.Imaging.PixelFormat.Format32bppRgb</code> can be used for this purpose. (Inherited from TestObject)
ClearText	Clears the textfield's text
Click	Clicks on the object. (Inherited from IClickable)
DomClick	Invokes a click using the DOM API. Alternatively you can invoke the Click function which will use low level replay for DomElements and subclasses. (Inherited from DomElement)
DomDoubleClick	Invokes a double click using the DOM API. Alternatively you can invoke the DoubleClick function which will use low level replay for DomElements and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)

Name	Description
DomMouseMove	Invokes a mouse move using the DOM API. Alternatively you can use <code>MouseMove</code> function which will use low level replay for <code>DomElements</code> and subclasses. This method is not supported for mobile web applications. (Inherited from DomElement)
DoubleClick	Double-clicks a mouse button on the object. (Inherited from IClickable)
ExecuteJavaScript	Evaluates the given JavaScript code within the parent document's context. (Inherited from DomElement)
Exists	Checks if any child object matching the locator exists for an object in the application under test. If the <i>timeout</i> parameter is passed the agent retries finding until the given timeout expires. If no timeout is specified and no object is found initially <i>false</i> is returned immediately. (Inherited from TestObject)
Exists	Checks if an object exists in the application under test. (Inherited from TestObject)
Find	Finds an object specified by an XPath locator. (Inherited from TestObject)
FindAll	Finds all objects specified by an XPath locator. (Inherited from TestObject)
GenerateLocator	Returns a locator for this object. (Inherited from TestObject)
GetChildren	Returns the child objects of this object. (Inherited from TestObject)
GetCssStyle	Retrieves the computed CSS style with the specified style name. (Inherited from DomElement)
GetDomAttribute	Gets the value of an object specific DOM attribute. (Inherited from DomElement)
GetDomAttributeList	Returns the names of all DOM attributes that are available for this element. This method is not supported for mobile web applications. (Inherited from DomElement)
GetDynamicMethodList	Returns a list of methods (including their signature) that can be dynamically invoked on this <code>TestObject</code> . (Inherited from TestObject)
GetHtmlRect	Retrieves the rectangle of a <code>DomElement</code> relative to the <code>BrowserWindow</code> in HTML pixels. In contrast to the screen coordinates that are provided by the <code>GetRect</code> function, the coordinates provided by <code>GetHtmlRect</code> are not affected by the zoom level on mobile browsers. The <code>click</code> function also uses HTML pixel coordinates for all <code>DomElements</code> . We recommend to use <code>GetHtmlRect</code> for all position calculations within the browser. This will ensure that your scripts are cross-browser capable and your position calculations are not effected by changing zoom levels on mobile browsers. (Inherited from DomElement)
GetParent	Looks up the parent of this object in the test application. (Inherited from TestObject)

Name	Description
GetProperty	Returns the value of the specified property. (Inherited from TestObject)
GetPropertyList	Returns a list of property names that can be retrieved for the given object. (Inherited from TestObject)
GetRect	Returns the size and position of this object. Per default the returned coordinates are relative to the parent window that contains this object. (Inherited from TestObject)
GetText	Gets the visible text of a DOM element. (Inherited from DomElement)
Highlight	Highlights a DOM element. (Inherited from DomElement)
HighlightObject	Highlights this object. (Inherited from TestObject)
ImageClick	Clicks on specified image asset. (Inherited from TestObject)
ImageClickFile	Clicks on the specified image. (Inherited from TestObject)
ImageExists	Returns whether the specified image asset exists. (Inherited from TestObject)
ImageExistsFile	Returns whether the specified image exists. (Inherited from TestObject)
ImageRectangle	Returns the object-relative rectangle of the specified image asset. (Inherited from TestObject)
ImageRectangleFile	Returns the object-relative rectangle of the specified image. (Inherited from TestObject)
Invoke	Dynamically invokes a method on the test object. (Inherited from TestObject)
InvokeMethods	Dynamically invokes a sequence of methods starting at this TestObject. (Inherited from TestObject)
MouseMove	Moves the pointer to the specified location in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
PressKeys	Presses (but does not release) a set of keys or mouse buttons. (Inherited from IKeyable)
PressMouse	Presses (but does not release) a mouse button in the object. This method is not supported for mobile Web applications. (Inherited from IClickable)
ReleaseKeys	Releases a set of keys or mouse buttons. (Inherited from IKeyable)
ReleaseMouse	Releases a mouse button. This method is not supported for mobile Web applications. (Inherited from IClickable)
ScrollIntoView	Scrolls the DOM element into the visible area of the browser window. (Inherited from DomElement)
SetDomAttribute	Sets the value of an object specific DOM attribute. (Inherited from DomElement)

Name	Description
SetFocus	Gives focus to the control. (Inherited from BrowserObject)
SetProperty	Sets the value of the specified property. (Inherited from TestObject)
SetText	Sets the text.
StartScreenRecording	Starts screen recording (Inherited from TestObject)
StopScreenRecording	Stops screen recording (Inherited from TestObject)
TextCapture	Returns the text in this object's visible area. (Inherited from TestObject)
TextClick	Clicks in the center of the specified text. (Inherited from TestObject)
TextExists	Returns whether the specified text exists. (Inherited from TestObject)
TextRectangle	Returns the object-relative rectangle of the specified text. (Inherited from TestObject)
TypeKeys	Sends a set of keystrokes to the object. (Inherited from IKeyable)
TypePasswordKeys	Types an encrypted password into an object, for example a text field. (Inherited from IKeyable)
Unhighlight	Restores the original foreground and background colors. (Inherited from DomElement)
WaitForChildDisappearance	Waits until the specified child object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForDisappearance	Waits until the object does not exist or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForObject	Waits for an object that matches the specified locator. If no object matches within an timeout an <code>ObjectNotFoundException</code> is thrown. The default timeout is 5 seconds and you can change the timeout by setting the value of the option <code>OPT_WAIT_RESOLVE_OBJDEF</code> . Use <code>WaitForObject</code> if the AUT takes a long time to display a specific object, for example when a transaction is processed before showing any results. By default, an action in the UI does not require a <code>WaitForObject</code> , because of the built-in synchronization. If you receive random timeout errors during normal script execution, consider increasing the default timeout instead of adding <code>WaitForObject</code> statements to your scripts. (Inherited from TestObject)
WaitForProperty	Waits until the specified property has the specified value or until the timeout is reached. If you don't explicitly pass a timeout

Name	Description
WaitForPropertyNotEquals	using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject) Waits until the specified property has a value that is different to the specified value or until the timeout is reached. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_RESOLVE_OBJDEF will be used. An exception is thrown if the timeout is reached. (Inherited from TestObject)
WaitForScreenshotStable	Waits until the control is visually stable and does not change its position, by comparing screenshots taken with the specified interval. If you don't explicitly pass a timeout using the optional timeout parameter, the timeout specified by the option OPT_WAIT_SCREENSHOT_STABLE will be used. The default interval is 200 milliseconds. An exception is thrown if the timeout is reached. (Inherited from TestObject)

IItemSelectable Interface

Description

A ItemSelectable provides a list of items of which one can be selected.

Syntax

'Declaration

```
Public Class IItemSelectable
```

Properties

Name	Description
ItemCount	the number of items in the listbox.
Items	the list of items in the listbox.

Methods

Name	Description
Select	Selects an item.

索引

記号

.NET スクリプト

バージョン、比較 240

比較 240

コア クラス 239

相違点のマージ 241

バージョン、相違点のマージ 241

引数のエクスポート、eCATT スクリプト 478

引数を eCATT スクリプトからインポートする 478

.NET のサポート

Silverlight 458

Windows Forms の概要 444

Windows Presentation Foundation (WPF) の概要
449

概要 444

Java Network Launching Protocol (JNLP)

スクリプトの記録 117

外部データ

テスト、使用 136

キーワード駆動テスト

アクティブ データの使用 136

Silk Central キーワードの実装 543

アプリケーション構成 532

インポートとエクスポート 313

概要 528

キーワードレコメンド、アルゴリズム 538

キーワード、検索 546

キーワードのアップロード、Silk Central 544

キーワードの削除 535

キーワードの実装 532

キーワードの追加 535

基本状態 532

記録 530

クラス リファレンス 1811

作成 530

作成、アセット ブラウザ 307

テスト メソッドの指定 534

トラブルシューティング 547

名前の変更 309

並べ替え、アセット ブラウザ 311

パラメータ、例 539

ビジュアル テストの指定 533

開く 307

フィルタリング、アセット ブラウザ 310

編集 535

変数を指定した実行 541

利点 528

既存のスクリプトの実行

iOS 9.3 421

機能

iOS 341, 420

サイズの指定

ブラウザー ウィンドウ 509

詳細設定

オプション 592

証明書

置換、Information Service 342

インストール、UFT Mobile 425

スクリプトの実行

停止 279

セキュアな接続

Information Service 341, 420

前提条件

Android、ネイティブ モバイル アプリケーション 403

Android、モバイル Web アプリケーション 403

Apple Safari 495

Google Chrome 500

iOS、ネイティブ モバイル アプリケーション 410

iOS、モバイル Web アプリケーション 410

属性除外リスト

MSUIA、設定 576, 587

UI オートメーション、設定 576, 587

設定 577

属性のタイプ

UI オートメーション 251, 523

停止

スクリプトの実行 279

データの追加

テスト 136

テキスト解決

概要 254

テキスト クリックの記録

概要 254

ポートの構成

Information Service、クライアント 340

Open Agent 339

メソッドの動的呼び出し

Apache Flex 371

Apache Flex カスタム コントロール 375

Java AWT 397, 401

Java Swing 397, 401

Java SWT 397, 401

SAP 466

SAP コントロール 467

Silverlight 460

UI オートメーション 249

Windows Forms 445

Windows Presentation Foundation (WPF) 452

モバイル Web アプリケーション

Android、前提条件 403

iOS、前提条件 410

制限事項 439

呼び出し

Android 442

iOS 442

UI オートメーション 249

A

AbstractButton class 2273

Access

データベース、コンパクト化 613

データベースのレコードのロック解除 613

- データベースバージョンの更新 612
- データベースを開く 611
- AccessibleControl class 2706
- ActiveData クラス 814
- ActiveDataRow クラス 819
- AddColumn メソッド
 - ActiveData 815
- AddRow メソッド
 - ActiveData 815
- Adobe Flex
 - Adobe AIR のサポート 381
 - automationName プロパティ 389
 - FlexDataGrid コントロール 382
 - Select メソッド、概要 381
 - Select メソッド、設定 391
 - アプリケーションの作成 387
 - 構成情報の追加 385
 - コンテナ 392
 - コンテナのコーディング 392
 - 実行時のパラメーター渡し 386
 - 実行時の読み込み 384
 - 実行時読み込み 385
 - 実行前のパラメーター渡し 386
 - セキュリティ設定 395
 - テストの再生 394
 - パラメータを渡す 386
 - 複数ビュー コンテナ 393
 - メソッドの呼び出し 371
- Agent クラス
 - コア クラス 825
- AJAX アプリケーション
 - 記録オプション、設定 577
 - スクリプトのハング 515
- Android
 - メソッドの呼び出し 442
 - モバイル Web アプリケーション、前提条件 403
 - モバイル ネイティブ アプリケーション、前提条件 403
 - USB デバッグの有効化 405
 - USB ドライバのインストール 405
 - エミュレータを設定する 406
 - 推奨設定 406
 - テスト 403
 - デバイスの開放 430
 - デバイスの開放、記録 431
 - デバイスの開放、再生 431
 - トラブルシューティング 432
 - ハイブリッド アプリケーション 404
 - 並列テスト、テスト済みの構成 407
- Android エミュレータ
 - 設定 406
- Apache Flex
 - automationIndex プロパティ 388
 - automationName プロパティ 388
 - Component Explorer 370
 - Flash Player 設定 369
 - Select メソッド、設定 391
 - アプリケーションの事前コンパイル 383
 - アプリケーションの有効化 382
 - オートメーション パッケージのリンク 383
 - 概要 369
 - カスタム コントロール 260, 370
 - カスタム コントロール、実装 376
 - カスタム コントロール、定義 372, 379, 388
 - カスタム コントロールのメソッドの呼び出し 375
 - 記録、アプリケーション 393
 - クラス定義ファイル 379, 388
 - コントロールが認識されない 396
 - 初期化、アプリケーション 393
 - スクリプトのカスタマイズ 380
 - スタイル 394
 - 属性 395, 520
 - テスト 370
 - 複数のアプリケーションのテスト 380
 - メソッドの呼び出し 371
 - ワークフロー 393
- Apache Flex アプリケーション
 - カスタム属性 325, 388
- API 再生
 - 比較、ネイティブ再生 489
- Apple Safari
 - 制限事項 497
 - Information Service、インストール 416, 421, 496, 499
 - サポート 483
 - 準備 495
 - 接続文字列 492
 - 前提条件 495
 - テスト 495
 - 複数のテストの実行 499
- Apple Safari のテスト
 - Information Service、インストール 416, 496
- Applet class 2277
- AppletContainer class 2281
- ArgumentAttribute クラス
 - キーワード駆動テスト クラス 1811
- Attach メソッド
 - Agent 826
- AWTButton class 2285
- AWTCanvas class 2289
- AWTCheckbox class 2293
- AWTCheckboxMenuItem class 2297
- AWTChoice class 2301
- AWTComponent class 2305
- AWTContainer class 2309
- AWTDialog class 2314
- AWTFrame class 2318
- AWTHorizontalScrollbar class 2323
- AWTLabel class 2327
- AWTList class 2332
- AWTMenu class 2336
- AWTMenuItem class 2340
- AWTMenuItem class 2343
- AWTRadioButton class 2347
- AWTScrollbar class 2351
- AWTScrollPane class 2355
- AWTTextArea class 2360
- AWTTextComponent class 2364
- AWTTextField class 2369
- AWTVerticalScrollbar class 2374
- AWTWindow class 2378

B

BaseGuiTestObject class 663
BaseState クラス
 コア クラス 831, 834
BasicArrowButton class 2383
BLOB
 eCATT 472
BLOB でデータを転送する
 スタンドアロン Silk Test Workbench を構成する 470
BrowserApplication class 2784
BrowserObject class 2790
BrowserWindow class 2793

C

CBanner class 1705
CEF
 テスト 481
CharSet enumeration 667
CheckBox class 667
CheckBoxToolItem class 672
CheckedListBox class 2739
Chrome
 既知の問題 654
 拡張、テスト 501
 クロス ブラウザー スクリプト 517
 構成設定 490
 前提条件 500
 テスト 499
 ユーザー データ ディレクトリ、テスト 501
 ロケーター 517
Chrome for Android
 サポート 483
 ブラウザの種類、設定 438
Chromium Embedded Framework
 テスト 481
ClickType
 data type 853
CloseAll メソッド
 ConsoleWindow 836
Close メソッド
 ConsoleWindow 836
Color
 data type 853
ColumnCount プロパティ
 ActiveData 819
ComboBox class 675
CommandLineArguments プロパティ
 BaseState 832
CommandLinePattern プロパティ
 BaseState 832
Common Class Reference 663
Common プロジェクト
 資産、接頭部の追加 570
 説明 35, 615
Component Explorer
 Apache Flex 370
Configuration Assistant
 自動サイン 417
configuring database connection 608
Connect メソッド
 Agent 827

ConsoleWindow クラス
 コア クラス 835
Control class 680
CoolBar class 1709
CoolItem class 1713
Count プロパティ
 ActiveData 819, 824
CTabFolder class 1716
CTabItem class 1721
Customer Care 76, 77

D

data types
 about 853
 ClickType 853
 Color 853
 ItemIdentifier 854
 ItemPath 855
 ModifierKeys 855
 MouseButton 856
 Point 856
 Range 857
 Rectangle 857
 TextPosition 858
 TextRange 858
 TreeContent 859
 TreeNode 859
database connection
 configuring 608
DataGrid class 2743
DataGridColumn class 2747
DataGridItem class 2751
DataGridRow class 2755
Decrypt メソッド
 Agent 827
Desktop class 687
Desktop プロパティ
 Agent 830
DetachAll メソッド
 Agent 828
Dialog class 687
Dispose メソッド
 ActiveData 816
 ActiveDataRow 820
dll
 Visual Basic からの呼び出し 241
 Visual Basic スクリプトからの呼び出し 242
 関数の宣言構文 242
 関数への引数の受け渡し 243
 関数への文字列引数の受け渡し 245
 規則の変更 246
 名前のエイリアス設定 245
 変更可能な引数の関数への受け渡し 244
 呼び出しの例 243
DllCall クラス
 コア クラス 839
DLL の呼び出し
 Visual Basic 241
 Visual Basic スクリプト 242
 例 243

- DLL ファイル
 - スクリプトでの参照 239
- DomainUpDown class 2759
- DomButton class 2797
- DomCheckBox class 2802
- DomDataList class 2806
- DomElement class 2811
- DomEmbeddedElement class 2816
- DomForm class 2820
- DomLink class 2825
- DomListBox class 2829
- DomRadioButton class 2834
- DomTable class 2838
- DomTableRow class 2843
- DomTextField class 2847
- DropDownToolItem class 691
- DSN
 - Access 605
 - Oracle 607
 - SQL Server 606
 - 概要 605
- DynamicInvoke
 - Android 442
 - Apache Flex 371
 - iOS 442
 - Java AWT 397, 401
 - Java Swing 397, 401
 - SAP 466
 - Silverlight 460
 - UI オートメーション 249
 - Windows Forms 445
 - Windows Presentation Foundation (WPF) 452

E

- eCATT
 - BLOB 472
 - BLOB 追跡情報の入力 473
 - BLOB の内容の入力または変更 473
 - eCATT UI からスクリプトを記録する 471
 - eCATT UI からスクリプトをテストする 471
 - eCATT UI からスクリプトを記録/再生する 472
 - eCATT スクリプトから .NET スクリプトへの引数のインポート 478
 - eCATT スクリプトからビジュアル テストへの引数のインポート 476
 - eCATT スクリプト引数の作成 475
 - eCATT スクリプト引数のサポート 474
 - eCATT でのテスト スクリプト属性の指定 471
 - RFC 接続パラメータの設定または変更 470
 - SAP/R3 ログイン パラメータの設定 470
 - Silk Test Workbench テスト結果 479
 - Silk Test Workbench テストを実行する 479
 - Silk Test Workbench と統合する 468
 - Silk Test Workbench を登録する 469
 - 概要 468
 - スクリプト引数の編集 475
 - スタンドアロン Silk Test Workbench を使用した eCATT スクリプトの作成 472
 - テスト スクリプトの定義 471
 - 統合オプション 568

- 統合のセットアップ 468
- 引数コンテナからの引数の削除 476
- 引数のエクスポート、.NET スクリプトから eCATT スクリプト 478
- ビジュアル テストでの eCATT スクリプトへの引数のエクスポート 477
- ユーザー アカウントを作成する 469
- eCATT と統合する
 - 概要 468
- ECCUST_ET
 - Silk Test Workbench を登録する 469
- Eclipse
 - トラブルシューティング 402
- Eclipse RCP
 - サポート 399
- Edge
 - 既知の問題 657
 - 制限事項 507
 - 接続文字列 492
 - テスト 507
 - リモート テスト 492
 - ロケーター 517
- Elapsed プロパティ
 - Timer 844
- ElapsedMilliseconds プロパティ
 - Timer 844
- ElementHost class 2763
- Encrypt メソッド
 - Agent 828
- Executable プロパティ
 - BaseState 833
- ExecutablePattern プロパティ
 - BaseState 832
- Execute メソッド
 - BaseState 841
- ExecuteBaseState メソッド
 - Agent 828
- ExecutionMode enumeration 695
- ExecutionResult class 696
- ExpandBar class 1725
- ExpandItem class 1729
- ExternalRowNumber プロパティ
 - ActiveDataRow 824

F

- FAQ
 - オブジェクト マップ 361
 - クロス ブラウザ テスト 513
 - コードの追加、AUT 259
 - 動的呼び出し 257
- FileHandle class 696
- FileInfo class 696
- FileOpenMode enumeration 697
- FilePointerMode enumeration 697
- FileShareMode enumeration 698
- FileSizeUnit enumeration 698
- Firefox
 - 制限事項 505, 506
 - 拡張機能、テスト 505
 - クロス ブラウザー スクリプト 517

- 構成設定 490
- テスト 503
- プロファイル、テスト 504
- ロケーター 517
- Flash Player
 - アプリケーションを開く 369
 - セキュリティ設定 395
- Flex
 - Adobe AIR のサポート 381
 - automationIndex プロパティ 388
 - automationName プロパティ 388, 389
 - Component Explorer 370
 - Flash Player 設定 369
 - FlexDataGrid コントロール 382
 - Select メソッド、概要 381
 - Select メソッド、設定 391
 - アプリケーションの作成 387
 - アプリケーションの事前コンパイル 383
 - アプリケーションの有効化 382
 - オートメーション パッケージのリンク 383
 - 概要 369
 - カスタム コントロール 260, 370
 - カスタム コントロール、実装 376
 - カスタム コントロール、定義 372, 379, 388
 - カスタム コントロールのメソッドの呼び出し 375
 - 記録、アプリケーション 393
 - クラス定義ファイル 379, 388
 - 構成情報の追加 385
 - コンテナ 392
 - 実行時のパラメーター渡し 386
 - 実行時の読み込み 384
 - 実行時読み込み 385
 - 実行前のパラメーター渡し 386
 - 初期化、アプリケーション 393
 - スクリプトのカスタマイズ 380
 - スタイル 394
 - セキュリティ設定 395
 - 属性 395, 520
 - テスト 370
 - テストの再生 394
 - パラメータを渡す 386
 - 複数のアプリケーションのテスト 380
 - 複数ビュー コンテナ 393
 - メソッドの呼び出し 371
 - ワークフロー 393
- Flex Class Reference 859
- FlexAccordion class 860
- FlexAdvancedDataGrid class 867
- FlexAlert class 876
- FlexApplication class 883
- FlexAreaChart class 889
- FlexAreaSeries class 896
- FlexAxisRenderer class 902
- FlexBarChart class 908
- FlexBarSeries class 916
- FlexBox class 922
- FlexBubbleChart class 928
- FlexBubbleSeries class 936
- FlexButton class 942
- FlexButtonBar class 948
- FlexCandlestickChart class 955
- FlexCandlestickSeries class 962
- FlexCanvas class 968
- FlexCartesianChart class 975
- FlexChart class 982
- FlexChartLegend class 988
- FlexChartSeries class 994
- FlexCheckBox class 1000
- FlexColorPicker class 1006
- FlexColumnChart class 1013
- FlexColumnSeries class 1020
- FlexComboBase class 1026
- FlexComboBox class 1033
- FlexContainer class 1040
- FlexContainerMovieClip class 1046
- FlexDataGrid class 1052
- FlexDateChooser class 1060
- FlexDateField class 1066
- FlexDisplayObject class 1074
- FlexDividedBox class 1078
- FlexForm class 1085
- FlexFormItem class 1091
- FlexHLOCChart class 1098
- FlexHLOCSeries class 1105
- FlexHLOCSeriesBase class 1111
- FlexImage class 1117
- FlexLabel class 1123
- FlexLineChart class 1128
- FlexLineSeries class 1135
- FlexLinkBar class 1142
- FlexList class 1148
- FlexListBase class 1156
- FlexListLabel class 1163
- FlexLoader class 1168
- FlexMenu class 1173
- FlexMenuBar class 1179
- FlexNavigationBar class 1185
- FlexNumericStepper class 1192
- FlexObject class 1198
- FlexOLAPDataGrid class 1203
- FlexPanel class 1213
- FlexPieChart class 1220
- FlexPieSeries class 1226
- FlexPlotChart class 1233
- FlexPlotSeries class 1240
- FlexPopUpButton class 1246
- FlexProgressBar class 1252
- FlexRadioButton class 1258
- FlexRepeater class 1264
- FlexRule class 1268
- FlexScrollBar class 1273
- FlexScrollBase class 1279
- FlexSlider class 1284
- FlexStandalonePlayer class 1290
- FlexTabNavigator class 1295
- FlexTextArea class 1302
- FlexTitleWindow class 1308
- FlexToggleButtonBar class 1315
- FlexTree class 1322
- FlexUIMovieClip class 1330
- FlexVideoDisplay class 1335
- FlexViewStack class 1340

FlexWindow class 1347
FlexWindowedApplication class 1354
FormatDate 関数
 式デザイン 183
FormatTime 関数
 式デザイン 185
FormsHost class 2767
FormsWindow class 2771

G

GetContents メソッド
 ConsoleWindow 837
GetDouble メソッド
 ActiveDataRow 820
GetEnumerator メソッド
 ActiveData 817
GetLong メソッド
 ActiveDataRow 821
GetLongLong メソッド
 ActiveDataRow 821
GetOption メソッド
 Agent 828
GetString メソッド
 ActiveDataRow 822
Google Chrome
 既知の問題 654
 制限事項、macOS 503
 iframe パフォーマンス、向上 511
 macOS 493
 オプション、設定 493
 拡張、テスト 501
 ケイパビリティ、設定 493
 構成設定 490
 サポート 483
 制限事項 501
 全画面表示 514
 前提条件 500
 追加のバージョン、テスト 512
 テスト 499
 ユーザー データ ディレクトリ、テスト 501
 リモート テスト 492
Group class 699
GUI
 概要 38
 メイン画面 39
GuiTestObject class 703
GWT
 コントロールの検索 324

H

Header class 2710
HorizontalSash class 1733
HorizontalScrollBar class 706
HTTPS
 証明書、置換 342
 Information Service 341, 420
 証明書、UFT Mobile 425

I

IAWTScrollable interface 2387

IAWTScroller interface 2388
IBaseScrollable interface 2388
IBaseState
 インターフェイス 840
IClickable interface 711
IFocusable interface 711
iframe
 パフォーマンス、向上 511
IItemSelectable interface 2852
IKeyable interface 711
IME
 UI オートメーション 253
IMobileClickable interface 1813
IMobileGestures interface 1814
IMobileKeyable interface 1814
IMoveable interface 712
INativeWindow interface 713
Information Service
 証明書、置換 342
 編集 341, 420
 ポート、設定 338
 ポートの構成、クライアント 340
 HTTPS 341, 420
 Mac、インストール 416, 421, 496, 499
innerHTML
 xBrowser 516
innerText
 xBrowser 516
IntelliSense
 使用法 113
Internet Explorer
 link.select のフォーカスの問題 515
 既知の問題 656
 クロス ブラウザー スクリプト 517
 構成設定 490
 サポート 483
 スクリプト例の起動 119
 全画面表示 514
 例、新しいブラウザ ウィンドウ 117
 ロケーター 517
Internet Explorer 10
 予期しない Click 動作 517
invoke
 Java AWT 397, 401
 Java SWT 397, 401
 SAP 466
 Silverlight 460
 Swing 397, 401
 Windows Forms 445
 Windows Presentation Foundation (WPF) 452
InvokeMethods
 Android 442
 Apache Flex 371
 iOS 442
 Java AWT 397, 401
 Java Swing 397, 401
 SAP 466
 Silverlight 460
 UI オートメーション 249
 Windows Forms 445
 Windows Presentation Foundation (WPF) 452

Invoke メソッド
 呼び出し可能なメソッド 257
 IOracleFormsMenuBase interface 2389
 IOracleFormsScrollable interface 2389
 IOracleFormsScroller interface 2390
 iOS
 テスト、デベロッパ アカウントなし 418
 デバイス、準備 415
 メソッドの呼び出し 442
 モバイル Web アプリケーション、前提条件 410
 モバイル ネイティブ アプリケーション、前提条件 410
 Information Service、インストール 416, 421, 496, 499
 Mac、準備 417
 Web アプリ、シミュレータ 413
 Web アプリ、テスト 413
 アプリ、テストの準備 415
 推奨設定 421
 テスト 409
 デバイスの開放 430
 デバイスの開放、記録 431
 デバイスの開放、再生 431
 ネイティブ アプリ、シミュレータ 412
 ネイティブ アプリ、テスト 411
 ハイブリッド アプリケーション 414
 iOS 9.3
 既存のスク립ト、実行 421
 ISapContextMenuable interface 1860
 IScrollable interface 713
 IsRunning プロパティ
 Timer 845
 Item class 713
 ItemIdentifier
 data type 854
 ItemPath
 data type 855
 Item メソッド
 ActiveData 817

J
 Java AWT
 概要 396
 属性 397, 521
 属性の種類 397, 521
 メソッドの呼び出し 397, 401
 Java AWT and Swing Class Reference 2272
 Java FX
 サポート 247
 Java Network Launching Protocol (JNLP)
 ビジュアル テスト 83
 Java Swing
 概要 396
 属性 397, 521
 メソッドの呼び出し 397, 401
 Java SWT
 サポート 399
 属性の種類 400, 521
 トラブルシューティング 402
 メソッドの呼び出し 397, 401
 Java SWT Class Reference 1705

 Java SWT クラス リファレンス 400
 Java AWT/Swing
 priorlabel 398
 JButton class 2391
 JCheckBox class 2395
 JCheckBoxMenuItem class 2399
 JColorChooser class 2404
 JComboBox class 2408
 JComponent class 2412
 JDesktopPane class 2417
 JDialog class 2421
 JEditorPane class 2425
 JFrame class 2430
 JHorizontalScrollBar class 2435
 JLabel class 2439
 JLayeredPane class 2443
 JList class 2447
 JMenu class 2452
 JMenuBar class 2456
 JMenuItem class 2460
 JNLP
 スク립トの記録 117
 ビジュアル テスト 83
 JPanel class 2465
 JPasswordField class 2469
 JPopupMenu class 2473
 JProgressBar class 2477
 JRadioButton class 2482
 JRadioButtonMenuItem class 2486
 JRootPane class 2490
 JScrollBar class 2494
 JScrollPane class 2499
 JSlider class 2503
 JSpinner class 2507
 JSplitPane class 2512
 JTabbedPane class 2516
 JTable class 2520
 JTableHeader class 2525
 JTextArea class 2529
 JTextComponent class 2534
 JTextField class 2539
 JTextPane class 2543
 JToggleButton class 2548
 JToolBar class 2552
 JTree class 2557
 JVerticalScrollBar class 2561
 JViewport class 2566
 JWindow class 2570

K
 key mnemonics
 Rumba 1835
 KeywordAttribute クラス
 キーワード駆動テスト クラス 1812
 KeywordGroupAttribute クラス
 キーワード駆動テスト クラス 1813

L
 Label class 717
 Link class 721

ListBox class 725
ListView class 2714
LoadActiveData メソッド
 Workbench 846
LoadAssembly
 アセンブリをコピーできない 259
Locator プロパティ
 BaseState 833
logging on 26, 608

M

Mac
 Apple Safari、前提条件 495
 Apple Safari、テスト 495
 Information Service、インストール 416, 421, 496,
 499
Menu class 729
MenuItem class 733
MenuStrip class 2776
MFC
 サポート 482
Microsoft Edge
 既知の問題 657
 制限事項 507
 iframe パフォーマンス、向上 511
 サポート 483
 接続文字列 492
 追加のバージョン、テスト 512
 テスト 507
 リモートテスト 492
Microsoft Foundation Class
 サポート 482
Microsoft UI オートメーション
 テストの記録 248
Microsoft ユーザー補助
 オブジェクト解決の向上 246
Mobile Class Reference 1813
MobileButton class 1815
MobileDevice class 1819
MobileObject class 1823
MobileTextField class 1827
MobileWindow class 1831
ModifierKeys
 data type 855
MonthCalendar class 2718
MouseButton
 data type 856
Mozilla Firefox
 制限事項 505, 506
 iframe パフォーマンス、向上 511
 macOS 493
 オプション、設定 493
 拡張機能、テスト 505
 既知の問題 658
 ケイバビリティ、設定 493
 構成設定 490
 サポート 483
 全画面表示 514
 追加のバージョン、テスト 512
 テスト 503

プロファイル、テスト 504
リモートテスト 492
MSUIA
 オプション、設定 576, 587
 メソッドの呼び出し 249
 有効化 576, 587
 オブジェクト解決、改善 247

N

Name プロパティ
 Timer 845
NumericUpDown class 2780

O

Open Agent
 接続ポート、構成 339
 ポートの構成、Information Service 340
 ポートの設定、リモート エージェント 338
 ポート番号 338
 オプション 329
 概要 329
 起動 329
 スクリプトから開始 329
 スクリプトから停止 329
 テスト、リモート 343
Open Agent の開始
 スクリプト 329
Open Agent の停止
 スクリプト 329
Open メソッド
 ConsoleWindow 838
OPT_ENABLE_ACCESSIBILITY
 オプション 592
OPT_ENABLE_EMBEDDED_CHROME_SUPPORT
 オプション 481
OPT_ENABLE_MOBILE_WEBVIEW_FALLBACK_SUPPORT
 オプション 592
OPT_IMAGE_ASSET_DEFAULT_ACCURACY
 オプション 592
OPT_IMAGE_VERIFICATION_DEFAULT_ACCURACY
 オプション 592
OPT_LOCATOR_ATTRIBUTES_CASE_SENSITIVE
 オプション 592
OPT_REMOVE_FOCUS_ON_CAPTURE_TEXT
 オプション 592
Oracle
 ORA_DBA グループの作成 601
 Oracle 認証の設定 600
 OS 認証 599
 概要 598
 クライアント接続用のデータベースの準備 601
 クライアントの設定 601
 データベース、コピー 612
 データベースの準備 604
 データベースの新規作成 600
 データベースのレコードのロック解除 613
 データベース バージョンの更新 612
 データベースを開く 611
 ドメインなしでの設定 604, 605

認証方式 598
パスワード認証スキーマ 603
ユーザー、設定 599
ユーザーに対するサーバー アクセスの作成 602
ユーザーの設定 599, 602

Oracle Forms
サポートするバージョン 398
前提条件 399
属性 399
について 398
OracleFormsApplication class 2574
OracleFormsButton class 2579
OracleFormsCheckbox class 2583
OracleFormsChoice class 2587
OracleFormsComboBox class 2591
OracleFormsContainer class 2596
OracleFormsHorizontalScrollbar class 2600
OracleFormsLabel class 2605
OracleFormsListBox class 2609
OracleFormsListView class 2614
OracleFormsMenu class 2620
OracleFormsMenuItem class 2624
OracleFormsPopList class 2628
OracleFormsPopupMenu class 2633
OracleFormsRadioButton class 2637
OracleFormsScrollbar class 2641
OracleFormsStatusArea class 2646
OracleFormsStatusBar class 2650
OracleFormsStatusBarItem class 2654
OracleFormsStatusIndicator class 2658
OracleFormsTabBar class 2662
OracleFormsTabBarItem class 2666
OracleFormsTabPanel class 2670
OracleFormsTextField class 2674
OracleFormsTitleBar class 2679
OracleFormsToolBar class 2683
OracleFormsToolBarItem class 2687
OracleFormsTree class 2692
OracleFormsVerticalScrollbar class 2697

P

Pager class 2723
Pause メソッド
Timer 842
Point
data type 856
priorlabel
Java AWT/Swing テクノロジ ドメイン 398
priorLabel
Win32 テクノロジ ドメイン 481
Private 変数
スクリプト 221
ProgressBar class 2727
Public 変数
スクリプト 221
PushButton class 737
PushToolItem class 741

Q

QT
サポート 247

R

RadioList class 745
RadioListToolItem class 749
Range
data type 857
Rectangle
data type 857
RegistryCategory enumeration 753
RegistryView enumeration 753
RemoveColumn メソッド
ActiveData 817
RemoveRow メソッド
ActiveData 817
RenameColumn メソッド
ActiveData 818
Reset メソッド
ActiveData 818
ResetOptions メソッド
Agent 829
ResultComment メソッド
Workbench 852
Resume メソッド
Timer 843
RowNumber プロパティ
ActiveDataRow 824
Rumba
key mnemonics 1835
RumbaKey class 1835
Unix ディスプレイ 465
画面検証の使用 464
サポートの有効化と無効化 463
について 463
ロケーター属性 464, 524
Rumba Class Reference 1835
Rumba data types
RumbaCharacterAttribute 1838
Rumba ロケーター属性
コントロールの識別 464, 524
RumbaCharacterAttribute
Rumba data types 1838
RumbaField class 1839
RumbaLabel class 1843
RumbaObject class 1847
RumbaScreen class 1851
RumbaTextField class 1855
RunScript メソッド
Workbench 849

S

Safari
制限事項 497
準備 495
接続文字列 492
前提条件 495
テスト 495
複数のテストの実行 499

SAP

概要 465
既知の問題 658

- クラス リファレンス 465
- セキュリティ設定 468
- 属性の種類 465, 521
- メソッドの呼び出し 466
- SAP Class Reference 1860
- SapBarChart class 1860
- SapBox class 1865
- SapButton class 1869
- SapCalendar class 1873
- SapChart class 1877
- SapCheckBox class 1881
- SapColorSelector class 1885
- SapComboBox class 1890
- SapComponent class 1894
- SapContainer class 1898
- SapContainerShell class 1902
- SapContextMenu class 1906
- SapCustomControl class 1909
- SapDockShell class 1913
- SapGridView class 1917
- SapHorizontalScrollBar class 1924
- SapHTMLViewer class 1927
- SapLabel class 1932
- SapMenu class 1936
- SapMenubar class 1940
- SapNetPlan class 1943
- SapOfficeIntegration class 1948
- SapOkCodeField class 1952
- SapPicture class 1956
- SapRadioButton class 1960
- SapScrollbar class 1964
- SapScrollContainer class 1967
- SapShell class 1971
- SapSimpleContainer class 1975
- SapSplitterContainer class 1979
- SapStatusbar class 1983
- SapTab class 1987
- SapTable class 1991
- SapTabStrip class 1995
- SapTextEdit class 1999
- SapTextField class 2004
- SapTitlebar class 2008
- SapToolbar class 2012
- SapToolbarControl class 2016
- SapTree class 2020
- SapUserArea class 2027
- SapVerticalScrollBar class 2031
- SapWindow class 2034
- SAP コントロール
 - メソッドの動的呼び出し 467
- Sash class 1737
- SashForm class 1741
- Sauce Labs
 - 有効化 426
- Save メソッド
 - ActiveData 818
- Scale class 754
- ScrollableControl class 1745
- ScrollBar class 758
- ScrolledComposite class 1749
- Select メソッド
 - Apache Flex、設定 391
 - SeparatorItem class 762
 - SetDouble メソッド
 - ActiveDataRow 822
 - SetLocation メソッド
 - Android 432
 - SetLong メソッド
 - ActiveDataRow 822
 - SetLongLong メソッド
 - ActiveDataRow 823
 - SetOption メソッド
 - Agent 829
 - SetString メソッド
 - ActiveDataRow 823
 - Shell class 1753
 - Shutdown メソッド
 - Agent 830
 - Silk Central
 - Sauce Labs、有効化 426
 - UFT Mobile、有効化 424
 - キーワードのアップロード 544
 - 場所の設定 542
 - パラメータ 205
 - Silk Central キーワード
 - 実装 543
 - Silk Performer
 - 実行時間の計測 264
 - Silk Test Workbench
 - 機能 30
 - 基本機能 27
 - 詳細 26
 - Silverlight
 - 概要 458
 - クラス リファレンス 458
 - サポート 458
 - スクロール 461
 - 属性の種類 458, 522
 - トラブルシューティング 462
 - メソッドの呼び出し 460
 - ロケーター属性 458, 522
 - Silverlight Class Reference 2039
 - SLApplication class 2039
 - SLAutoCompleteBox class 2043
 - SLBase class 2048
 - SLButton class 2053
 - SLCalendar class 2056
 - SLCalendarButton class 2061
 - SLCalendarDayButton class 2065
 - SLCheckBox class 2068
 - SLComboBox class 2072
 - SLComboBoxItem class 2077
 - SLDataGrid class 2081
 - SLDataGridCell class 2086
 - SLDataGridDetails class 2089
 - SLDataGridRow class 2093
 - SLDataPager class 2097
 - SLDatePicker class 2101
 - SLDescriptionViewer class 2105
 - SLFrame class 2109
 - SLGridSplitter class 2113
 - SLGroup class 2117

SLHeader class 2120
 SLHeaderItem class 2124
 SLHorizontalScrollBar class 2128
 SLHyperlinkButton class 2132
 SLImage class 2136
 SLListBox class 2140
 SLListItem class 2144
 SLMediaElement class 2148
 SLMenu class 2152
 SLMenuBar class 2156
 SLMenuItem class 2160
 SLMultiScaleImage class 2163
 SLPane class 2167
 SLPasswordBox class 2171
 SLPopup class 2175
 SLProgressBar class 2179
 SLRadioButton class 2183
 SLRepeatButton class 2187
 SLRichTextBox class 2190
 SLSeparator class 2194
 SLSlider class 2198
 SLSpinner class 2202
 SLSplitButton class 2206
 SLStatusBar class 2210
 SLTabControl class 2214
 SLTabItem class 2218
 SLTable class 2222
 SLTextBlock class 2225
 SLTextBox class 2229
 SLThumb class 2233
 SLTitleBar class 2237
 SLToggleButton class 2241
 SLToolBar class 2245
 SLToolTip class 2248
 SLTreeView class 2252
 SLTreeViewItem class 2257
 SLValidationSummary class 2261
 SLVerticalScrollBar class 2265
 SLWindow class 2269
 SparkAirHTML class 1362
 SparkApplication class 1367
 SparkBorderContainer class 1373
 SparkButton class 1379
 SparkButtonBar class 1385
 SparkButtonBarButton class 1391
 SparkButtonBase class 1397
 SparkCheckBox class 1402
 SparkComboBox class 1408
 SparkComplexDisplay class 1415
 SparkDataGrid class 1422
 SparkDataGridLabel class 1428
 SparkDataGroup class 1433
 SparkDataRenderer class 1440
 SparkDropDownList class 1446
 SparkDropDownListBase class 1453
 SparkForm class 1460
 SparkFormItem class 1466
 SparkGroup class 1472
 SparkGroupBase class 1479
 SparkImage class 1485
 SparkLabel class 1491
 SparkList class 1496
 SparkListBase class 1503
 SparkListLabel class 1509
 SparkMuteButton class 1516
 SparkNavigatorContent class 1521
 SparkNumericStepper class 1527
 SparkObject class 1533
 SparkPanel class 1538
 SparkPopUpAnchor class 1544
 SparkRadioButton class 1550
 SparkRange class 1556
 SparkRichEditableText class 1561
 SparkRichText class 1567
 SparkScrollBar class 1573
 SparkSkinnableContainer class 1579
 SparkSkinnableContainerBase class 1585
 SparkSkinnableDataContainer class 1590
 SparkSkinnablePopUpContainer class 1596
 SparkSkinnableTextBase class 1602
 SparkSlider class 1608
 SparkSpinner class 1614
 SparkTabBar class 1619
 SparkTextArea class 1626
 SparkTextBase class 1632
 SparkTextInput class 1638
 SparkTileGroup class 1644
 SparkTitleWindow class 1651
 SparkToggleButton class 1657
 SparkToggleButtonBase class 1663
 SparkTrackBase class 1668
 SparkVideoDisplay class 1674
 SparkVideoPlayer class 1679
 SparkVolumeBar class 1685
 SparkWindow class 1690
 SparkWindowedApplication class 1697
 Spinner class 1757
 SplitPaneDivider class 2702
 SQL Server
 SQL Server データベースの新規作成 596
 管理ユーザー、作成 596
 構成する 595
 データベース、コピー 612
 データベースの準備 598
 データベースのレコードのロック解除 613
 データベース バージョンの更新 612
 データベースを開く 610
 ドメインなしでの設定 604
 ユーザーの設定 597
 Start メソッド
 Timer 843
 StartNew メソッド
 Timer 843
 StatusBar class 2731
 Stop メソッド
 Timer 844
 STW.exe
 ログ 566
 STW.EXE
 前提条件、Silk Central 558
 パラメータ 548
 例 558

- STW.EXE、前提条件 558
- StyledText class 1761
- SupportLine 76, 77
- Swing
 - 概要 396
 - 属性 397, 521
 - メソッドの呼び出し 397, 401
- SWTBrowser class 1766
- SWTDateTime class 1770
- SWTTabControl class 1774
- SWTTabItem class 1778
- SWTTable class 1782
- SWTTableColumn class 1786
- SWTTableRow class 1790
- SWTTree class 1794
- SWTTreeColumn class 1799
- SystemFunctions class 684
- SystemFunctions プロパティ
 - Agent 830

T

- TabControl class 766
- Table class 770
- TableColumn class 774
- TableRow class 778
- TechDomains プロパティ
 - BaseState 833
- TestObject class 782
- textContent
 - xBrowser 516
- TextField class 784
- TextPosition
 - data type 858
- TextRange
 - data type 858
- Timer クラス
 - コア クラス 842
- ToggleButton class 789
- ToolBar class 793
- ToolItem class 797
- Tree class 800
- TreeContent
 - data type 859
- TreeNode
 - data type 859
- TypeKeys メソッド
 - ConsoleWindow 838

U

- UAC
 - 概要 126
- UFT Mobile
 - 証明書、インストール 425
 - 有効化 423
 - 有効化、Silk Central 424
- UI
 - 概要 38
- UI オートメーション
 - オプション、設定 576, 587

- 制限事項 253
- 属性のタイプ 251, 523
- メソッドの呼び出し 249
- 有効化 576, 587
- ローケータ属性 251, 523
- オブジェクト解決、改善 247
- スクロール 252
- テストの記録 248
- トラブルシューティング 253
- Unicode コンテンツ
 - サポート 254
- Unicode 文字のマッピング 612
- Unix ディスプレイ
 - Rumba 465
- UnloadAllDllsFromAgent メソッド
 - DllCall 840
- UnloadDllFromAgent メソッド
 - DllCall 840
- UpDown class 2734
- URL プロパティ
 - BrowserBaseState 835
- USB ドライバのインストール
 - Android 405
- Username プロパティ
 - Workbench 852
- UWP
 - トラブルシューティング 480
- UWP アプリ
 - サポート 479

V

- Verify メソッド
 - Workbench 849
- VerifyAsset メソッド
 - Workbench クラス 851
- VerticalSash class 1803
- VerticalScrollBar class 805
- ViewForm class 1807
- Visual Basic
 - DLL の呼び出し 241
 - スクリプトからの DLL の呼び出し 242
- Visual COBOL
 - サポートするバージョン 462
 - について 462

W

- Web ページ
 - キャプチャ、全画面 297
- WebSync 76, 77
- Web アプリケーション
 - xBrowser テスト オブジェクト 486
 - カスタム属性 326, 519
 - カスタム属性、設定 577
 - 既知の問題 654
 - 記録オプション、設定 577
 - サポートされている属性 518, 524
 - 設定 585
- Win32
 - priorLabel 481
- Win32 Class Reference 2706

- Win32 クラス リファレンス 481
- Window class 809
- Windows
 - 属性の種類 450, 525
- Windows 8
 - 制限事項 520
- Windows 8.1
 - 制限事項 520
- Windows API
 - サポート 480
- Windows Forms
 - 概要 444
 - クラス リファレンス 444
 - 属性の種類 445, 525
 - メソッドの呼び出し 445
- Windows Forms Class Reference 2739
- Windows Presentation Foundation (WPF)
 - WPFIItemsControl クラス 451
 - 概要 449
 - カスタム コントロール 452
 - クラスの公開 457, 579
 - クラス リファレンス 450
 - メソッドの呼び出し 452
 - ロケーター属性 450, 525
- Windows Forms アプリケーション
 - カスタム属性 327, 445
- WinForms アプリケーション
 - カスタム属性 327, 445
- Workbench クラス 845
- WorkingDirectory プロパティ
 - BaseState 834
- Works Order 番号 76, 77
- WPF
 - WPFIItemsControl クラス 451
 - カスタム コントロール 452
 - クラスの公開 457, 579
 - クラス リファレンス 450
 - メソッドの呼び出し 452
 - ロケーター属性 450, 525
- WPF アプリケーション
 - カスタム属性 327, 451
- WPF ロケーター属性
 - コントロールの識別 450, 525

X

- xBrowser
 - Internet Explorer で四角形の位置が正しくない 515
 - 機能の公開 515
 - 認識されないダイアログ 513
- xBrowser
 - Apple Safari 495
 - Chrome for Android、設定 438
 - DomClick が Click のように動作しない 514
 - FAQ 513
 - FieldInputField.DomClick でダイアログが開かない 514
 - Google Chrome 499
 - innerHTML 516
 - innerText 516
 - innerText がロケーターで使用されない 515
 - link.select のフォーカスの問題 515
 - Microsoft Edge 507
 - Mozilla Firefox 503
 - textContent 516
 - 新しいページへの移動 515
 - オブジェクト解決 487
 - オブジェクト マップ、使用 347
 - 概要 483
 - カスタム属性、設定 577
 - クラス リファレンス 520
 - クロス ブラウザー スクリプト 517
 - 現在のブラウザの種類、表示 514
 - 再生、API とネイティブの比較 489
 - スクロール 514
 - 設定 585
 - 属性の種類 518, 524
 - 正しくないタイムスタンプ、ログ 515
 - 正しくないロケーターの記録 515
 - テスト オブジェクト 486
 - フォント タイプの検証 515
 - ブラウザ構成設定 490
 - ページ同期 488
 - マウス移動の記録 516
 - マウス移動の詳細設定、設定 490, 574
 - ロケーターにないクラスとスタイル 517
 - ロケーターの記録 517
- xBrowser Class Reference 2784
- xBrowser テスト
 - Apple Safari、制限事項 497
 - Microsoft Edge、制限事項 507
 - 現在のブラウザの種類、表示 514
 - ブラウザの記録オプション、設定 577
- XPath
 - トラブルシューティング 327

あ

- アイコン
 - テスト ステップ 641
 - 表示、メッセージ ボックス 99
- アイコンの表示
 - メッセージ ボックス 99
- アクセス
 - データベース、コピー 612
- アクティブ オブジェクト
 - 記録中のハイライト 134
- アクティブ プロジェクト
 - 説明 35, 615
- アクティブ データ
 - 概要 136
 - キーワード駆動テスト 136
 - 資産 137
 - 資産の作成 137, 150
 - 利点 137
 - インポートとエクスポート 313
 - オプション、詳細設定 593
 - 行数、ビジュアル テスト 149
 - 繰り返しロジック、作成 60
 - 繰り返しロジック、ステップの定義 60
 - 繰り返しロジック、データのマップ 61
 - 繰り返しロジックの作成、ビジュアル テスト 144

- 検索する 311
- 更新されたデータの保存、ビジュアル テスト 148
- 再生、ビジュアル テスト 62
- 削除 310
- シートの選択、VB .NET スクリプト 153
- シートの選択、ビジュアル テスト 149
- 資産の作成、チュートリアル 59
- 使用、ビジュアル テスト 143
- 使用するデータの指定 141
- スクリプト概要 150
- スクリプトへの組み込み 151
- チュートリアル 58
- データの更新、式の使用 147
- データの更新、自動保存の回避 148
- データの更新、ステップ テキスト 146
- データの更新、ビジュアル テスト 147
- データの準備 138
- データのマップ、ステップ テキスト 146
- データのマップ、ビジュアル テスト 145
- データ ファイル、確認 59
- データ ファイル、作成 139
- データ ファイル、準備 138
- 名前の変更 309
- 並べ替え 311
- 表示、複数バージョン 312
- 開く 307
- ファイルの編集 141
- フィルタリング 310
- プロパティ 99
- 分析結果、ビジュアル テスト 62
- 読み取りオプションの設定、データ ファイル 140
- 例、詳細 92
- 例、テストシナリオ 91
- 例、ビジュアル テスト 91
- 列数、ビジュアル テスト 149
- アクティブ データ資産
 - 関連付け、ビジュアル テスト 143
 - 作成、アセット ブラウザ 307
- アクティブ データの更新
 - 自動保存の回避 148
- アクティブ データの保存
 - 自動保存の回避 148
- アセット ブラウザ
 - 概要 647
 - 検索する 311
 - 資産の作成 307
 - 資産の並べ替え 311
 - 資産を開く 307
 - ビジュアル テストを開く 89
 - フィルタリング 310
- 値の選択
 - 条件 174
- 新しいブラウザ ウィンドウ
 - ビジュアル テスト 84
- アップロードする
 - キーワード ライブラリ 544
 - ライブラリ 544
- アドオン
 - Google Chrome 501
 - Mozilla Firefox 505
- アプリケーション構成

- Java Network Launching Protocol (JNLP)、スクリプト 117
- Java Network Launching Protocol (JNLP)、ビジュアル テスト 83
- エラー 127
- 環境変数 128
- キーワード駆動テスト 532
- 削除 123
- 定義 122
- テクノロジ ドメインの無効化 126
- トラブルシューティング 128
- 変更 123
- 無応答のアプリケーションのキャンセル 586
- アプリケーション準備完了タイムアウト
 - 設定 586
- アプリケーションの選択
 - ダイアログ ボックス 125
- アプリのアップロード
 - Mac 403
- 暗号化
 - パスワード 265
- 安定した識別子
 - について 322
- 安定したロケーター
 - 作成 323
- 安定したロケーターを作成する
 - 概要 323

い

- 一時停止
 - タイマー 213
 - ビジュアル テスト 213
- 移動
 - 資産、プロジェクト間 309
 - 指定したラベル、ビジュアル テスト 210
 - テスト ステップ 210
 - ビジュアル テストでのラベル 207
 - ラベル テスト ステップの作成、ビジュアル テスト 209
- イメージ クリック
 - イメージ資産変数 365
 - 記録 362
- イメージ解決
 - 概要 362
 - 記録設定の設定 573
 - メソッド 362
 - 有効化 362
- イメージ クリックの記録
 - 概要 362
- イメージ検証
 - イメージ資産変数 367
 - 概要 365
 - 記録中に追加する 367
 - 作成 365
 - 失敗時の相違点の表示 367
 - [相違点] ビュー 368
 - ビジュアル テストへの追加 366
- イメージ資産
 - 作成 363
 - インポートとエクスポート 313

- 概要 363
- 作成、アセットブラウザ 307
- 詳細設定の設定 573
- フィルタリング、アセット ブラウザ 310
- 複数のイメージ、追加 364
- イメージのクリック
 - 記録の設定 573
- イメージのチェック
 - 概要 365
- 印刷
 - スクリプト 275
 - スクリプトの結果 295
 - ビジュアル テスト 270
 - ビジュアル テストの結果 294
 - ヘルプ トピック 76
- インストール
 - Information Service、Mac 416, 421, 496, 499
- インストールする
 - 必要な権限 595
- インポート
 - 資産 313
- インポートの権限
 - 資産 316

う

- [ウィンドウ] メニュー 634
- 埋め込み Chrome
 - テスト 481
- 上書きする
 - オプション、再生 282

え

- エイリアス
 - ロケータの使用 573
- エージェント
 - ポートの構成、Information Service 340
 - ポート番号 338
 - オプション 329
 - 概要 329
 - 起動 329
 - クラス リファレンス 814
- エージェント オプション
 - Open Agent 329
- エクスポートする
 - 資産 315
- エクスポートの権限
 - 資産 316
- エミュレータ
 - 定義、再生 422
 - テスト 403
- エラー
 - 概要 55
 - 確認 58
 - 自動処理の設定、プロパティ ペイン 269
 - 出力ウィンドウに表示 651
 - 処理、再生 268
 - 調べる 55
 - データベース 615
 - デバッグ 56
- エラー時の移動先プロパティ
 - プロパティ 104

- エラー処理
 - ロジック 168
- エラー処理ロジック
 - ビジュアル テスト、追加 168
- エラーのトレース
 - スクリプト 293
 - ビジュアル テスト 293
- エンコード
 - パスワード 265
- 演算子
 - 式デザイナ 179

お

- オブジェクト
 - .NET スクリプトを識別する 118
 - オブジェクト識別ダイアログ、ビジュアル テスト 86
 - 解決、タイムアウトの設定 586
 - 検索する 317
 - 識別、画面プレビュー 86
 - 識別、テスト対象アプリケーション 85
 - 存在確認 321
 - プロパティ 104
- オブジェクト タイプ
 - ロケーター 317
- オブジェクト マップ エントリ
 - 名前の変更 347
- オブジェクト解決
 - Exists メソッド 321
 - FindAll メソッド 322
 - Find メソッドの使用 321
 - UWP、トラブルシューティング 480
 - 安定したロケーターを作成する 322
 - 改善、UI オートメーション 247
 - 概要 317
 - カスタム属性 325
 - 属性の使用 318
 - 複数のオブジェクトの識別 322
 - ユーザー補助を使用して向上する 246
- オブジェクト解決タイムアウト
 - 設定 586
- オブジェクト解決の向上
 - ユーザー補助 246
- オブジェクトの解決
 - xBrowser 487
- オブジェクトの識別
 - 概要 317
 - ダイアログ ボックス 648
- [オブジェクトの識別] ダイアログ
 - ビジュアル テスト 86
- オブジェクトの待機
 - 時間の指定 570
 - ビジュアル テスト 211
- オブジェクトの同期化
 - 再生遅延の設定 213
 - ビジュアル テスト 211
- オブジェクト マップ
 - バージョン、比較 351
 - 比較 351
 - FAQ 361
 - Web アプリケーション 347

- xBrowser 347
- インポートとエクスポート 313
- オフに切り替え 345
- オンに切り替え 345
- 概要 344
- 記録 345, 573
- 検索する 311
- 項目のグループ化 360
- 項目のコピー 354
- 項目の削除 359
- 項目の追加 355
- 項目の名前変更 347
- コピー 309
- 削除 310
- 削除、スクリプト 361
- 作成、アセットブラウザ 307
- 手動作成 361
- 詳細設定の設定 573
- スクリプトから開く 356
- スクリプトの構文 111
- スコープ 345, 573
- 相違点のマージ 352
- 操作の記録中のマージ 346
- 名前の変更 309
- 並べ替え 311
- バージョン、相違点のマージ 352
- ビジュアルテストでのロケータからオブジェクト
マップへの移動 83
- 表示、複数バージョン 312
- 開く 307
- フィルタリング 310
- ベストプラクティス 359
- 変更 349
- マージ 361
- 利点 345
- ロック 573
- オブジェクト マップ項目
 - エラーの検出 358
 - グループ化 360
 - コピー 354
 - 削除 359
 - 識別 350, 357
 - 追加 355
 - テストアプリケーションからの更新 352
 - テストアプリケーションでの検索 357
 - ハイライト 357
 - ロケータの変更 350
- オブジェクト マップの記録
 - 有効化/無効化 573
- オプション
 - .NET スクリプトの再生結果 590
 - MSUIA、設定 576, 587
 - UI オートメーション、設定 576, 587
 - 詳細設定 592
 - eCATT の統合 568
 - OPT_ENABLE_EMBEDDED_CHROME_SUPPORT 481
 - アクティブ データ、詳細設定 593
 - 開始画面 571
 - 概要 568
 - 記録 572, 579, 582

- 記録、保存の適用 583
- 記録の出力、ビジュアル テスト 581
- グローバル 568
- 再生 583
- 再生、値の取得 282
- 再生、上書き 281, 282
- 再生、保存の適用 591
- 再生オプションの上書き 282
- 再生結果、キーワード駆動テスト 590
- 再生結果、設定 588
- 再生ステータス ダイアログ ボックス、設定 300
- 再生のタイミング、設定 586
- スクリプト作成 592
- 全般 569
- 全般再生、設定 584
- 統合 568
- ビジュアル テストの再生 281
- ビジュアル テストの再生結果 589
- ブラウザの記録、設定 577
- 保存 582, 591
- オプション プロファイル
 - 記録、適用 583
 - 再生用の作成 591
 - 適用 591
- オフに切り替え
 - 基本状態 129
- 親スクリプト
 - ドライバ スクリプト 112
- オンに切り替え
 - 基本状態 129

か

- 開始画面
 - オプション 571
 - 概要 39, 625
 - ビジュアル テスト
 - 記録、開始画面 80
 - ビジュアル テスト、記録 80
 - フラグ 571
- 開始行
 - アクティブ データ 141
- 開始タイマー
 - ビジュアル テスト 213
- 概要
 - Oracle データベースの構成 598
 - 自動テスト 32
 - テスト戦略 32
 - 利点 27
- 拡張
 - Google Chrome 501
 - VB .NET スクリプト 218
 - ビジュアル テスト、概要 49
- 拡張機能
 - Mozilla Firefox 505
- カスタマイズ
 - [ツール] メニュー 633
 - ビジュアル ナビゲータのレイアウト 646
- カスタム コントロール
 - Apache Flex の動的呼び出し 375
 - Apache Flex、実装 376

- Apache Flex、定義 379, 388
- AUT にコードを追加する 257
- AUT へのコードの追加、FAQ 259
- Flex、定義 372
- 概要 256
- カスタム クラスの作成 262
- 管理 261
- サポート 262
- 挿入したコードが AUT で使用されない 259
- ダイアログ ボックス 263
- テストする (Apache Flex) 260, 370
- 動的呼び出し、FAQ 257
- 呼び出しで予期しない文字列が返される 257
- カスタム コントロールのテスト
 - AUT にコードを追加する 257
- カスタム属性
 - UI オートメーション、設定 576, 587
 - Apache Flex アプリケーション 325, 388
 - Web アプリケーション 326, 519
 - Windows Forms アプリケーション 327, 445
 - WPF アプリケーション 327, 451
 - 記録オプション、設定 577
 - コントロール 325
 - 設定 578
- カスタム プロパティ
 - コントロール 325
- 画面キャプチャ
 - .NET スクリプトの再生オプション 590
 - 画面プレビュー 188
 - 記録オプション 579
 - 記録オプション、設定 581
 - 再生オプション、キーワード駆動テスト 590
 - ビジュアル テストの再生オプション 589
- 画面のキャプチャ
 - 手動 88
- 画面の最小化
 - 検索中 570
 - 再生中 570
- 画面のフラッシュ
 - 記録中 570
- 画面プレビュー
 - キャプチャされたイメージ、表示 643
 - コントロール、識別 86
 - コントロールの識別、コントロールの自動ステップ 186
 - ステップの削除 215
 - 設定、構成 188
 - ビジュアル テストの更新 49
 - 表示 644
- 環境変数
 - アプリケーション構成 128
- 監視
 - 再生中の変数 192
- 関数
 - 式デザイナー 181
- カンマ区切りファイル
 - アクティブ データ、準備 138
- 管理
 - 概要 595
 - キーワード 535
 - グループ 623

- グローバル変数 196
- 結果 291
- 資産 302
- スクリプト 235
- テスト フロー 207
- テスト フロー、移動の作成 210
- テスト フロー、ラベル テスト ステップの作成 209
- プロジェクト 35, 615
- プロジェクト、削除 620
- プロジェクト、追加 616
- プロジェクト、名前の変更 619, 620
- ユーザー 621
- ユーザー、削除 622
- ユーザー、追加 621
- ユーザー、編集 622
- ユーザー アクセス 621
- 管理者権限
 - インストールする 595
 - 実行 595
- 関連付け
 - アクティブ データ、ビジュアル テスト 145
 - アクティブ データ資産、ビジュアル テスト 143

き

- キーボード
 - ショートカット 36
- キーボード イベント遅延
 - 設定 586
- キーワード
 - プロジェクトでの検索 546
 - Silk Central にアップロードする 544
 - 管理 535
 - 記録 533
 - グループ化 546
 - 結合 535, 541
 - 削除 535
 - シーケンス 535
 - 実装 532, 535
 - 置換 535
 - 追加 535
 - データの受け渡し 534
 - テスト メソッドの指定 534
 - について 529
 - ネスト 535
 - パラメータ 535, 538
 - パラメータ、例 539
 - ビジュアル テスト、指定 533
 - 開く 535
 - 開く、Silk Central 542
 - フィルタリング 546
- キーワード シーケンス
 - インポートとエクスポート 313
 - 作成 541
 - 作成、アセット ブラウザ 307
 - 名前の変更 309
 - 並べ替え、アセット ブラウザ 311
 - パラメータ 538
 - 開く 307
 - フィルタリング、アセット ブラウザ 310
- キーワード ライブラリ

- アップロードする 544
- キーワード駆動
 - テスト 528
- キーワード駆動テスト エディター
 - 推薦するキーワード 538
- キーワード駆動テストの実行
 - 変数 541
- キーワードの削除
 - キーワード駆動テスト 535
- キーワードの追加
 - キーワード駆動テスト 535
- 既知の問題
 - Google Chrome 654
 - Internet Explorer 656
 - Microsoft Edge 657
 - Mozilla Firefox 658
 - SAP 658
 - Silk Test Workbench 659
 - Web アプリケーション 654
 - 一般的な問題 652
 - について 652
 - モバイル Web アプリケーション 654
- 基本機能
 - Silk Test Workbench 30
 - 自動テスト 32
 - ソフトウェア コンポーネント 32
 - テスト戦略 32
 - 利点 27
- 基本状態
 - 実行 129
 - オフに切り替え 129
 - オンに切り替え 129
 - キーワード駆動テスト 532
 - 詳細 128
 - 変更、スクリプト 132
 - 変更、ビジュアル テスト 132
 - 変更、ユーザー インターフェイス 130
- キャプチャ
 - Web ページ、全画面 297
 - アプリケーション画面、.NET スクリプト 590
 - アプリケーション画面、キーワード駆動テスト 590
 - アプリケーション画面、ビジュアル テスト 589
 - アプリケーションの画面、設定 581
- キャプチャ画面
 - 更新、再生 190
 - 更新、テスト アプリケーション 190
- キャプチャ画面の更新
 - 再生、ビジュアル テスト 190
 - すべて 190
 - ビジュアル テスト、テスト アプリケーション 190
- キャプチャされたイメージの表示
 - 画面プレビュー 643
- 行数
 - アクティブ データ、ビジュアル テスト 149
- 行の実行の制御
 - スクリプトのデバッグ 273
- 記録
 - Apache Flex アプリケーション 393
 - イメージ検証を追加する 367
 - オブジェクト マップ 345
 - オブジェクトのハイライト 134

- オプション 572, 582
- オプション プロファイル 582
- オプション プロファイル、適用 583
- 概要 134
- 画像が表示されない 432
- キーワード 533
- キーワード駆動テスト 530
- 既存のテストへの操作 121
- 事前読み込みの設定 457
- 出力オプション 579
- 出力オプション、ビジュアル テスト 581
- スクリプト、概要 67
- データ入力 220–223
- データベース 610
- デバイスの開放 431
- ビジュアル テスト 78, 88
- ビジュアル テスト、開始画面 80
- ビジュアル テスト、複数のアプリケーション 82
- ビジュアル テストを使用する利点 91
- モバイル アプリケーション 422
- 利用可能なアクション 134

<

- 組み込んだスクリプト
 - パラメータの使用、ビジュアル テスト 227
 - 変更する 208
- 組み込んだビジュアル テスト
 - 実行 277
 - パラメータ、値の割り当て 203
 - パラメータ、再利用 207
- クラス
 - 無視 575
- クラス リファレンス
 - エージェント 814
 - キーワード駆動テスト 1811
 - テスト オブジェクト 814
- 繰り返しロジック
 - アクティブ データ、ステップの定義 60
 - アクティブ データ、リテラル データへのマップ 61
 - アクティブ データ テスト 144
 - コンボ ボックスに基づく 161
 - 作成、アクティブ データ 60
 - ビジュアル テスト、追加 161
 - リストに基づく 161
- 繰り返しロジックの定義
 - コンボ ボックスに基づく 161
 - リストに基づく 161
- クリック
 - モバイル Web 443
- グループ
 - 管理 623
 - 権限の編集 624
 - 削除 624
 - 追加 623
- グループ化
 - オブジェクト マップ項目 360
 - キーワード 546
- クローズ オプション
 - 設定 588
- グローバル プロジェクト

- 定義 617
- グローバル参照
 - 定義 617
- グローバル変数
 - 管理 196
 - 作成、ビジュアル テスト 196
 - 取得、ビジュアル テスト 197
 - ビジュアル テスト 195
- クロス ブラウザ テスト
 - Apple Safari、制限事項 497
 - Microsoft Edge、制限事項 507
 - Apple Safari 495
 - FAQ 513
 - Google Chrome 499
 - Microsoft Edge 507
 - Mozilla Firefox 503
 - オブジェクト解決 487
 - オブジェクト マップ、使用 347
 - 概要 483
 - 現在のブラウザの種類、表示 514
 - スクロール 514
 - 接続文字列 492
 - 正しくないタイムスタンプ、ログ 515
 - テスト オブジェクト 486
 - ブラウザの記録オプション、設定 577
 - マウス移動の詳細設定、設定 490, 574
 - リモート ロケーション、追加 125
 - ロケータの記録 517

け

- 継続的インテグレーション
 - キーワード ライブラリのアップロード 544
- 結果
 - アセット ブラウザから開く 293
 - インポートとエクスポート 313
 - エラーの確認 58, 74
 - 概要 46
 - カスタマイズ 292
 - カスタマイズする 570
 - 管理 291
 - [結果] ウィンドウ 286, 646
 - 結果の記録のオン/オフの切り替え 292
 - 検索する 311
 - コピー 309
 - コメント 294
 - 最大数、設定 311
 - 削除 310
 - 作成 291
 - 作成、アセット ブラウザ 307
 - 実行ステータス 289
 - 自動的に開く 292
 - 手動で開く 292
 - スクリプトでの検証 219
 - スクリプトの印刷 295
 - ステップのステータス 289
 - タブ、ウィンドウ 287
 - ツールバー 286
 - データベース整合性チェック 615
 - 名前の変更 309
 - 並べ替え 311

- について 285
- ビジュアル テストへの切り替え 293
- ビジュアル テストの印刷 294
- ビジュアル ナビゲータ 290
- 表示、複数バージョン 312
- 開く 307
- ファイルの保存 296
- フィルタ 297
- フィルタ、作成 297
- フィルタ、適用 299
- フィルタの削除 300
- フィルタリング 310
- プロパティ 108
- 分析 276, 285
- [要約] タブ 287
- [結果] ウィンドウ
 - 概要 286, 646
 - 画面プレビュー 48
 - タブ 47
 - ツールバー 47
 - ビジュアル ナビゲータ 290
 - [プロパティ] ペイン 47
- 結果の作成
 - スクリプト 291
 - ビジュアル テスト 291
- 結果の分析
 - スクリプト 285
 - ビジュアル テスト 285
- 結果フィルタ
 - 適用 299
 - 複製 299
 - 編集 298
- 結合
 - キーワード 535
- 言語リファレンス
 - 概要 663
 - キーワード駆動テスト クラス 1811
 - コア クラス 814
- 現在の時刻の取得
 - FormatTime 関数 185
- 現在の日付を取得
 - FormatDate 関数 183
- 現在のブラウザの種類
 - 表示 514
- 検索する
 - キーワード、キーワード駆動テスト 546
 - 資産 311
- 検索範囲
 - ロケータ 317
- 検証
 - インポートとエクスポート 313
 - 作成、アセット ブラウザ 307
 - スクリプトへの追加 70, 219
 - 追加 50
 - データベース 615
 - フィルタリング、アセット ブラウザ 310
- 検証結果の説明
 - プロパティ 109
- 検証ロジック
 - 記録 78
 - 記録中のスクリプトの追加 219

コーディングによるスクリプトへの追加 219
追加する 171
ビジュアルテスト、追加 165

こ

コア クラス
 ConsoleWindow クラス 835
更新
 アクティブ データ ファイル、式の使用 147
 アプリケーションの画面 190
 データ、アクティブ データ ファイル 147
 データの保存、アクティブ データ ファイル 148
 ビジュアルテスト 87
構成する
 SQL Server 595
 再生、ビジュアルテスト 281
 ビジュアルテストの再生、挿入されたテスト 283
項目識別子
 ビジュアルテスト 191
項目の指定
 ビジュアルテスト 191
コード
 検索、VB .NET スクリプト 235
 [コード] ウィンドウ 650
 置換、VB .NET スクリプト 235
 ブックマークを使用したスクリプトでの移動 236
[コード] ウィンドウ
 について 650
子スクリプト
 ドライバ スクリプト 112
コピー
 結果フィルタ 299
 スクリプト 309
 データベース 612
 テスト ステップ 214
コマンド
 コマンド プロパティ 101
 ショートカットキー 36
 [ツール] メニューに追加 633
コマンド ライン
 silktest.exe コマンド ライン 548
 STW.EXE パラメータ 548
 STW.EXE の例 558
 STW エラー 555
 STW コマンド ラインの概要 548
 STW 出力ファイル 554
 資産インポートの構成ファイル 563
 資産エクスポートの構成ファイル 566
 資産のインポートとエクスポート 560
 資産のインポート パラメータ 561
 資産のエクスポート パラメータ 564
 入力ファイル 553
 ログ 566
コメント
 結果 294
 ビジュアルテストへの追加 153
 メッセージ ボックスでの表示 209
コントロール
 .NET スクリプトを識別する 118
 オブジェクト識別ダイアログ、ビジュアルテスト 86

識別、画面プレビュー 86
識別、テスト対象アプリケーション 85
待機、表示の安定 212
 ビジュアルテストでのプロパティ値 188
コントロールの検索
 GWT の例 324
 同列要素の例 323
コントロールの検証
 判断ロジックの利用、例 95
コントロールの識別
 コントロールの自動化、画面プレビュー 186
 コントロールの自動化、テスト アプリケーション 186
 コントロールの自動化、非表示コントロール 187
 動的ロケータ属性 527
コントロールの自動ステップ
 作成、画面プレビュー 186
 作成、テスト アプリケーションのコントロール 186
 作成、非表示コントロール 187
コンパクト化
 データベース、Access 613

さ

再開タイマー
 ビジュアルテスト 213
サイズ変更
 画面イメージ、画面プレビュー 644
再生
 .NET スクリプトの結果オプション 590
 エラー処理、ビジュアルテスト 268
 オブジェクトの待機 211
 オプション 583
 オプション、値の取得 282
 オプション プロファイル 591
 画面の最小化 570
 キャプチャ画面、更新 190
 記録されたスクリプト 69
 クローズ オプション 588
 結果オプション、キーワード駆動テスト 590
 結果オプション、設定 588
 再生ステータス ダイアログ ボックス、オプションの設
 定 300
 再生ステップのプロパティ 107
 事前読み込みの設定 457
 情報の表示 209
 スクリプト 279
 ステップ実行の一時停止 213
 すべてのファイルの保存 570
 設定 105
 設定、挿入されたスクリプト 283
 設定、挿入されたビジュアルテスト 283
 設定、ビジュアルテスト 281
 全般オプション、設定 584
 タイミング 213
 タイミング、オプションの設定 586
 テスト 276
 テスト ロジック デザインを使用して再生エラーに自動
 的に対処するようにビジュアルテストを設定
 する 269
 デバイスの開放 431
 デバイスの選択 422

- ビジュアル テスト 46
- ビジュアル テスト、組み込み 277
- ビジュアル テストでのスクリプトの実行 208, 226
- ビジュアル テストの結果オプション 589
- ブラウザの選択 484
- 保存したオプション、適用 591
- 再生
 - 認識されないダイアログ 513
- 再生エラー
 - 概要 55
 - 確認 58
 - 調べる 55
 - ダイアログ ボックス 301
 - デバッグ 56
 - 変数の追跡 57
- 再生オプション
 - 上書きする 281
- 再生結果
 - について 285
- 再生結果オプション
 - .NET スクリプト 590
 - キーワード駆動テスト 590
 - ビジュアル テスト 589
- 再生設定の取得
 - プロパティ 103
- 再生設定の設定
 - プロパティ 108
- 再生の完了
 - ダイアログ ボックス 301
- 再生の停止
 - スクリプト、選択したポイント 274
 - ビジュアル テスト、選択したポイント 266
- 再生ホット キー
 - 変更 591
- 再生モード
 - 設定 584
- 最大化
 - ブラウザ 514
- 削除
 - アプリケーション構成 123
 - キーワード 535
 - グループ 624
 - 結果フィルタ 300
 - 資産 310
 - 資産、古いバージョン 303
 - データベース レコード、Access 613
 - テスト ステップ、画面プレビュー 215
 - テスト ステップ、ストーリーボード 216
 - テスト ステップ、テスト ステップ ウィンドウ 215
 - プロジェクト 620
 - ユーザー 622
- 削除する
 - データベース レコード、Access 613
 - プロジェクト 620
 - ユーザー 622
- 作成
 - アクティブ データ資産 137, 150
 - DSN 605
 - アクティブ データ テスト 141
 - 移動、ビジュアル テスト 210
 - キーワード駆動テスト 530

- グローバル変数、ビジュアル テスト 196
- 結果フィルタ 297
- パラメータ 202
- ビジュアル テスト 78
- ビジュアル テスト、複数のアプリケーション 82
- プロジェクト 616
- ラベル テスト ステップ 209
- ラベル ステップ 207

参照

- スクリプトへの追加 239
- プロジェクト 616
- プロジェクト、グローバル 617
- ライブラリからの削除 240

サンプル アプリケーション

- 起動 42, 67
- 記録、ビジュアル テスト 43
- スクリプトの記録 68

し

- シートの選択
 - アクティブ データ、VB .NET スクリプト 153
 - アクティブ データ、ビジュアル テスト 149
- 式
 - アクティブ データの更新 147
 - 変更 178
- 式デザイン
 - Format Date 関数 183
 - FormatTime 関数 185
 - 演算子 179
 - 概要 176
 - 式の作成 176
 - 式の編集 178
- 式デザイナー
 - 関数 181
- 識別
 - オブジェクト、[オブジェクトの識別] ダイアログ ボックス 648
 - コントロール 186
 - コントロール、画面プレビュー 86
 - コントロール、テスト対象アプリケーション 85
- 識別子
 - 安定 322
- 識別する
 - コントロール、.NET スクリプト 118
- 時刻
 - FormatTime 関数 185
- 資産
 - インクルード元 312
 - 作成、アクティブ データ 137, 150
 - アクティブ データ 137
 - 移動 309
 - インクルード 312
 - インポート 313
 - インポートとエクスポート 313
 - インポートの権限 316
 - エクスポートする 315
 - エクスポートの権限 316
 - 概要 302
 - 管理 302
 - 結果、作成 291

- 削除 310
- 作成、アセットブラウザ 307
- 作成、命名規則 313
- サポートされる種類 302
- 資産のバージョンをパーズ 303
- スクリプトから開く 307, 365
- 接頭部の追加 570
- データ ファイル、作成 139
- デフォルトの保存動作、設定 311
- 名前の変更 309
- 並べ替え 311
- パーズするバージョンについて 303
- バージョン、相違点のマージ 308
- バージョン、比較 307
- 比較 307
- ビジュアル テスト、作成 78
- 表示、複数バージョン 312
- フィルタリング 310
- 複製 309
- マージ 308
- 命名規則 313
- 資産エクスポート ウィザード
 - 使用法 315
- 資産管理
 - 資産のバージョンの最大数 568
 - デフォルトの保存動作 568
- 資産の格納
 - 最大バージョン数、設定 311
 - バージョンの最大数 568
- 資産の作成
 - アセット ブラウザ 307
- 資産の種類
 - 利用可能 302
- 資産のバージョン
 - 最大数 568
 - 最大数、設定 311
- 資産のバージョンをパーズ
 - について 303
 - パラメータ、コマンド ライン 304
 - 方法 303
- 資産の表示
 - 複数バージョン 312
- 資産の保存
 - デフォルトの保存動作、設定 311
- 資産のメンテナンス
 - ユーザー、ログアウト 623
- 事前読み込み
 - 記録/再生時の設定 457
- 実行
 - スクリプト 279
 - スクリプト、概要 279
 - テスト 276
 - ビジュアル テスト 276
 - ビジュアル テスト、組み込み 277
 - ビジュアル テストでのターゲット アプリケーション
 - 210
- 実行ステータス
 - 結果 289
- 実行の遅延
 - テスト 264
- 実装
 - キーワード 535
- 自動
 - コントロール プロパティの使用 188
 - 操作の表示、再生 280
 - 操作の表示、設定 584
 - テスト ステップ 214
 - テスト ステップ、記録せずに作成 186
- 自動ステップ
 - 画面プレビュー、ビジュアル テスト 188
 - ステップの同期、ビジュアル テスト 188
- 自動保存の回避
 - アクティブ データ、ビジュアル テスト 148
- シミュレータ
 - 定義、再生 422
 - テスト 411
 - ネイティブ アプリ、テスト 412
 - モバイル Web アプリケーション、テスト 413
- 周辺機器が無い
 - テスト マシン 28
- 終了行
 - アクティブ データ 141
- 出カウィンドウ 651
- 出カパラメータ
 - スクリプト 237
 - スクリプトからの削除 238
 - 定義 65
 - 編集 238
- 手動作成
 - オブジェクト マップ 361
- 循環参照
 - 回避、プロジェクト 617
- 順序
 - アセット ブラウザの列 311
- 準備
 - Silk Test Workbench 26
- 条件
 - 値の選択 174
 - ダイアログ ボックス 300
- 条件デザイン
 - 値の選択 174
 - 条件の編集 174
- 条件デザイナー
 - 概要 171
 - 条件の定義 173
- 条件の選択
 - ダイアログ ボックス 174
- 条件の定義
 - プロパティに基づく 157, 161
- 詳細設定プロパティ
 - 概要 99
- 使用状況データの収集
 - 無効化 76
 - 有効化 76
- ショートカット
 - [デバッグ] メニュー 631
- ショートカット キー
 - リスト 36
- 除外される文字
 - 記録 134
 - 再生 134
- 初期化する

Apache Flex アプリケーション 393
シリアル番号 76, 77

す

推薦するキーワード

キーワード駆動テスト エディター 538

数値パラメータ

例 234

スクリーンキャスト

機能しない 432

スクリプト

.NET スクリプトでの参照 239

Java Network Launching Protocol (JNLP) の構成
117

再生 69, 71, 279

実行中の停止 279

バージョン、比較 240

比較 240

ロケーターの使用 320

DLL ファイルの参照 239

アクティブ データの使用 150, 151

新しいブラウザ ウィンドウ 117

アプリケーションの起動 119

印刷 275

インポート 313

インポートとエクスポート 313

エラー処理の例 273

エラーのトレース 293

オートメーション呼び出しの表示 584

オートメーション呼び出しの表示、再生 280

オブジェクト マッピング 344

拡張 218

確認 69

管理 235

記録 114

記録、概要 67

記録中の検証の追加 219

組み込んだ、変更 208

結果 285

結果、作成 291

結果の印刷 295

検索する 311

構文 111

コーディングによる検証の追加 219

コードの検索 235

コードの置換 235

コメントの挿入 274

再生、概要 279

再生エラー 74

再生の停止、選択したポイント 274

再利用 218

削除 310

作成 109

作成、アセット ブラウザ 307

作成、ベスト プラクティス 110

サンプル アプリケーション 42, 67

サンプル アプリケーションに対する記録 68

出力パラメータ 65

出力パラメーターの削除 238

出力パラメーターの追加 237

出力パラメーターの編集 238

手動作成 113

信頼性 272

スクリプトでの参照 239

スクリプトでのパラメーターの作成と受け渡し 228

スクリプト内からのスクリプトの参照 218

相違点のマージ 241

挿入 73

段階的実行、選択したポイントから 273

段階的実行、デバッグ モード 272

チュートリアル、2 つめのスクリプトの記録 73

テストをキーワードとして指定 534

デバッグ 271

デバッグ、行の実行の制御 273

ドライバ 112

ドライバーの作成 113

名前の変更 309

並べ替え 311

入力パラメータ 65, 236

入力パラメーターの削除 237

入力パラメーターの編集 236

バージョン、相違点のマージ 241

パラメータ、受け渡し 228

パラメータ、作成 228

パラメータ、定義 225

パラメータの受け渡し、スクリプト内 229

ビジュアル テストでの実行 208, 226

ビジュアル テストでの使用 65, 66

ビジュアル テストにデータを渡す 224

表示、複数バージョン 312

開く 121, 307

ファイルの削除 239

ファイルの追加 238

ファイル名の変更 239

フィルタリング 310

複数のアプリケーションの記録 116

複製 309

[プロパティ] ペインの表示 650

編集 121

変数 220–223

変数の宣言 221

保存 122

モジュール式 72

乱数の生成 63

利点 110

ロケーターの記録 120

スクリプト パラメータ

ビジュアル テスト、使用 227

スクリプト作成

オプション 592

スクリプト参照ライブラリ

参照の削除 240

スクリプトへの追加 239

スクリプト内からのスクリプトの参照 218

スクリプトの拡張

概要 70

検証の追加 70

再生 71

変数の追加 71

スクリプトの記録

サンプル アプリケーション 68

- チュートリアル、2 つめの記録 73
- スクリプトの再生
 - 拡張したスクリプト 71
 - モジュール式 74
- スクリプトの削除
 - オブジェクト マップ エントリ 361
- スクロール
 - クロス ブラウザ テスト 514
- スタイル
 - Flex アプリケーション 394
- ステップ
 - アクティブ データの更新 146
 - 実行の制御、再生 267
 - ビジュアル テスト 214
- ステップ実行の制御
 - デバッグ 267
- ステップの使用
 - 挿入 88
- ステップの選択
 - ダイアログ ボックス 170
- ステップの同期
 - ロジック ツールボックス 170
- ストーリーボード
 - 概要 645
 - サムネイルの移動 646
 - テスト ステップ、削除 216
- スプレッドシート
 - アクティブ データ、準備 138
- スリープ
 - テストへの追加 264

せ

- 制限事項
 - Apple Safari 497
 - Google Chrome 501
 - Google Chrome、macOS 503
 - Microsoft Edge 507
 - Mozilla Firefox 505
 - Windows 8 520
 - Windows 8.1 520
 - ネイティブ モバイル アプリケーション 440
 - モバイル Web アプリケーション 439
- 製品サポート 76, 77
- 製品スイート
 - コンポーネント 29
- セキュリティ設定
 - SAP 468
- 接続
 - データベース 607
- 接続の設定
 - データベース 608
- 接続文字列
 - デスクトップ ブラウザー、リモート 492
 - デスクトップ ブラウザー、ローカル 493
 - モバイル デバイス 426
- 接続ユーザー
 - データベース、表示 621
 - ログアウト 623
- 設定
 - アクティブ データ、詳細設定 593

- 記録 582
- 記録、保存の適用 583
- グループ 623
- 再生、全般 584
- 再生、保存の適用 591
- 再生結果、.NET スクリプト 590
- 再生結果、オプション 588
- 再生結果、キーワード駆動テスト 590
- 再生結果、ビジュアル テスト 589
- 再生ステータス ダイアログ ボックス 300
- 再生のクローズ オプションの設定 588
- 再生のタイミング 586
- 全般、変更 570
- ビジュアル テスト、新しいブラウザ ウィンドウ 84
- 変数のコントロール プロパティ値 188
- 保存 582, 591
- マウス移動の詳細設定、クロス ブラウザ テスト 490, 574
- ユーザー 621
- ユーザー、追加 621
- 設定を構成する
 - 画面プレビュー 188
- 全画面表示
 - ブラウザ 514
- 宣言
 - スクリプトにおける変数 221
- 選択
 - テスト ステップ、複数 214
- 全般
 - オプション 569
- 全般オプション
 - 変更する 570
- 全般プロパティ
 - 概要 102

そ

- 相違点
 - 資産、マージ 308
- 相違点の表示
 - イメージ検証 367
- [相違点] ビュー
 - イメージ検証 368
- 操作
 - 更新、テスト ステップ 190
 - テスト ステップ、設定 641
- 操作の記録
 - オブジェクト マップ エントリのマージ 346
- 操作の選択
 - ビジュアル テスト 153
- 操作の変更
 - ビジュアル テスト 153
- [操作] メニュー
 - コマンド 628
- 操作を記録する
 - 既存のテスト 121
 - ビジュアル テスト 87
- 挿入
 - 結果コメント 294
 - コメント 153
 - メニュー 630

挿入されたスクリプト
再生オプションの上書き 283
挿入されたビジュアル テスト
再生オプションの上書き 283

属性の種類
Apache Flex 395, 520
Java AWT 397, 521
Java Swing 397, 521
Java SWT 400, 521
Oracle Forms 399
SAP 465, 521
Silverlight 458, 522
Web アプリケーション 518, 524
Windows 450, 525
Windows Forms 445, 525
xBrowser 518, 524
概要 520

その他
プロパティ 103
ソフトウェア コンポーネント
概要 32

た

ターゲット アプリケーション
ビジュアル テストでの起動 210

ダイアログ
認識しない 513
ダイアログ ボックス
再生エラー 301
再生の完了 301
条件 300
条件の選択 174
ステップの選択 170
テストスクリプトの参照 240
ビジュアル テスト変数 278

待機
コントロール、表示の安定 212
プロパティ 109, 212

タイマー
プロパティ 108

タイミング
アプリケーションとの同期 211, 213
ステップ 213

タイムスタンプ
正しくない、クロス ブラウザ テスト 515

ダウンロード 76, 77
正しくないタイムスタンプ
ログ、クロス ブラウザ テスト 515

段階的実行
スクリプト、選択したポイント 273
スクリプト、デバッグ モード 272
ビジュアル テスト、デバッグ モード 268
ビジュアル テスト、選択したポイント 266

ち

遅延
キーボード イベント 586
ステップの実行 213
マウス イベント 586
遅延プロパティ
概要 102

置換
キーワード 535
チュートリアル
アクティブ データ 58
再生 66
乱数の生成 63

つ

追加
キーワード 535
グループ 623
資産、命名規則 313
[ツール] メニューの項目 633
テスト ステップ 207
プロジェクト 616
ユーザー 621
ローカル変数 193
ツールバー
操作 634
[ツール] メニュー
コマンド 632

て

定義
開く 122
停止タイマー
ビジュアル テスト 213
データ
アクティブ データ 141
外部、追加 139
共有、ビジュアル テスト 202
スクリプトでの再利用 220
ビジュアル テスト、再利用 191
データ型
変数、ビジュアル テスト 191
データ ソース名 (DSN)
作成 605
データ ソース名 (DSN) の作成
Access 605
データ ソース名 (DSN) の作成
Oracle 607
SQL Server 606
データの共有
ビジュアル テスト 202
データの再利用
グローバル変数 195
グローバル変数、作成 196
グローバル変数、取得 197
スクリプト パラメータ、ビジュアル テストでの使用
227
パラメータ、定義 225
パラメータ、ビジュアル テスト 207
ビジュアル テスト 191
ビジュアル テスト、変数 191
予約変数 197
予約変数、戻り値型 198
予約変数、リスト 197
ローカル変数 193
ローカル変数、編集 194

- データの種類
 - 例 222
- データの準備
 - アクティブ データ ファイル 138
- データベース
 - Access を開く 611
 - Oracle を開く 611
 - Oracle、作成 600
 - SQL を開く 610
 - 構成 595
 - 構成、共有 609
 - コピー 612
 - コンパクト化、Access 613
 - 整合性チェック 615
 - 接続 607
 - 接続、設定 608
 - 接続ユーザー、表示 621
 - バージョン、更新 614
 - バージョンの更新 612
 - 開く 610
 - 別の場所 33
 - 膨張の抑制 615
 - メンテナンス 610
 - メンテナンス、接続ユーザー 614
 - メンテナンス設定、構成 614
 - レコードのロック解除 613
- データベース メンテナンス
 - メッセージ、受信 614
- データベース接続
 - 設定 608
- データベースのメンテナンス
 - 設定、構成 614
- データベースのレコードのロック解除 613
- データベース レコード
 - 削除、Access 613
- テキスト
 - 表示 209
- テキストの表示
 - メッセージ ボックス 102
- テキスト ファイル
 - アクティブ データ、準備 138
- 適用
 - 記録オプション 583
 - 再生オプション 591
- テクノロジ ドメイン
 - 無効化 126
- デスクトップ警告
 - 表示 570
- テスト
 - 拡張 136
 - 再生 276
 - 作成 78
 - 実行の遅延 264
 - 操作を記録する 121
 - モジュール式 53, 72
 - 利点 27, 32
- テスト ステップ ウィンドウ
 - ステップの削除 215
- テスト マシン
 - 周辺機器が無い 28
- テスト メソッド
 - キーワードとして指定 534
- テスト ロジック
 - 概要 156
 - 繰り返しロジック 160
 - 繰り返しロジック テスト ステップの編集 164
 - 検証ロジック 164
 - 検証ロジック テスト ステップの編集 168
 - ステップのコピーおよび貼り付け 214
 - テスト ロジック デザイナ 169
 - テスト ロジック デザイナ ウィザード 171
 - 判断ロジック 156
 - 判断ロジック テスト ステップの編集 160
 - ロジック ツールボックス 170
- テスト ロジック デザイナ
 - [エラー処理の構築] ページ 168
 - 概要 169
 - 繰り返し、コンボ ボックスに基づく 161
 - 繰り返し、リストに基づく 161
 - [繰り返しの構築] ページ 161
 - [検証資産の構築] ページ 165
 - [使用するアクティブ データ資産の定義] ページ 161
 - プロパティに基づく条件、定義 157, 161
 - [ようこそ] ページ 169
 - ようこそページ、表示 570
 - 要約ページ、エラー処理ロジック 168
 - 要約ページ、繰り返しロジック 161
 - 要約ページ、検証ロジック 165
 - 要約ページ、判断ロジック 157
 - 要約ページ、表示 570
 - ロジック ステップ ペイン、表示 570
 - [ロジックの種類を選択 (エラー処理ロジック)] ページ 168
 - [ロジックの種類を選択 (繰り返しロジック)] ページ 161
 - [ロジックの種類を選択 (検証ロジック)] ページ 165
 - [ロジックの種類を選択 (判断ロジック)] ページ 157
- テスト ロジック デザイナ ウィザード
 - [繰り返し (While) の構築] ページ 161
 - [繰り返し回数の定義] ページ 161
 - [検証の構築] ページ 165
 - 挿入 171
 - 挿入する 171
 - [タイミングに基づく条件の定義] ページ 165
 - [判断の構築] ページ 157
- テスト アプリケーション
 - オブジェクト識別ダイアログ、ビジュアル テスト 86
 - 画面イメージ、表示 644
 - キャプチャ画面、更新 190
 - コントロールの識別 85, 186
 - コントロールの識別、コントロールの自動ステップ 186
 - コントロールを識別する、.NET スクリプト 118
 - 非表示コントロールの識別、コントロールの自動ステップ 187
- テスト オブジェクト
 - クラス リファレンス 814
- テスト 結果
 - 概要 46
 - 画面プレビュー 48
 - タブ 47
 - ツールバー 47

- [プロパティ] ペイン 47
- テスト自動化
 - 障壁 28
 - 同期 283
 - ビジュアル テスト 90
- テストスクリプト
 - 新しいブラウザ ウィンドウ 117
 - アプリケーションの起動 119
 - 記録 114
 - 結果 285
 - 結果、作成 291
 - 構文 111
 - 再生、概要 279
 - 作成 109
 - 作成、ベスト プラクティス 110
 - 手動作成 113
 - ドライバ 112
 - ドライバーの作成 113
 - 開く 121
 - 複数のアプリケーションの記録 116
 - 編集 121
 - 保存 122
 - 利点 110
 - ロケーターの記録 120
- テストスクリプトの参照
 - ダイアログ ボックス 240
- テストステップ
 - アイコン 641
 - アクティブ データの更新 146
 - アプリケーションの実行 210
 - 移動 207
 - 移動、指定したラベル 210
 - 画面の更新 190
 - 記録 87
 - 組み込んだスクリプト、変更 208
 - グローバル変数 195
 - グローバル変数、管理 196
 - グローバル変数、作成 196
 - グローバル変数、取得 197
 - コピー 214
 - 削除、画面プレビュー 215
 - 削除、ストーリーボード 216
 - 削除、テスト ステップ ウィンドウ 215
 - 実行の制御、再生 267
 - 自動、手動で作成 186
 - 種類、ビジュアル テスト 641
 - パラメータ、削除 205
 - パラメータ、作成 202
 - 貼り付け 214
 - 判断ロジック、例 95
 - 判断ロジック、例のシナリオ 95
 - 判断ロジック、例の詳細 96
 - ビジュアル テスト 214
 - ビジュアル ナビゲータ 639
 - 表示 85
 - 複数、選択 214
 - プロパティ、ビジュアル テスト 644
 - プロパティ、表示 644
 - 変数のコントロール プロパティ値 188
 - メッセージ ボックス 209
 - 予約変数 197

- 予約変数、戻り値型 198
- 予約変数、リスト 197
- ラベル 207
- ラベル、作成 209
- ローカル変数 193
- ローカル変数、追加 193
- ローカル変数、編集 194
- ローカル変数の削除 195
- テスト ステップの作成
 - ラベル 209
- テスト ステップの種類
 - ビジュアル テスト 641
- テストの記録
 - Microsoft UI オートメーション 248
- テストの再生
 - リモート マシン 343
- テスト フロー
 - 管理 207
- テストを実行する
 - eCATT 479
- デバイスが接続されていません
 - モバイル 432
- デバイスの開放
 - 記録 431
 - 再生 431
 - モバイル テスト 430
- デバッグ
 - エラー処理、ビジュアル テスト 268
 - 行の実行の制御、スクリプト 273
 - スクリプト 271
 - スクリプト、選択したポイントからの段階的実行 273
 - スクリプト、選択したポイントでの再生の停止 274
 - ステップ実行の制御、再生 267
 - 段階的実行、スクリプト 272
 - 段階的実行、ビジュアル テスト 268
 - テスト 266
 - ビジュアル テスト 266
 - ビジュアル テスト、選択したポイントからの段階的実行 266
 - ビジュアル テスト、選択したポイントでの再生の停止 266
 - ビジュアル テスト変数 192
- [デバッグ] メニュー
 - コマンド 631
 - ショートカット 631

と

- 同期
 - xBrowser 488
 - 設定の変更 283
 - 正しくないタイムスタンプ 515
 - について 283
- 同期オプション
 - xBrowser 585
- 統合
 - Silk Central の場所の設定 542
 - オプション 568
- 動的呼び出し
 - Android 442
 - AUT へのコードの追加、FAQ 259

- FAQ 257
- iOS 442
- 概要 256
- スクリプトの単純化 257
- 入力引数の型が一致しない 259
- モバイル ネイティブ 442
- 予期しない戻り値 257
- 動的ロケータ属性
 - 詳細 527
- 同列要素
 - 検索する 323
- ドキュメント
 - 配布物 34
 - 表記規則 75
- ドメインなしでの設定
 - Oracle の設定 605
 - SQL Server 604
- ドメインなしでのデータベースの設定
 - ユーザーの作成 604
- ドライバ スクリプト
 - 概要 112
 - 作成 113
- トラブルシューティング
 - Eclipse 402
 - Java SWT 402
 - Silverlight 462
 - UI オートメーション 253
 - XPath 327
 - アプリケーション構成 128
 - キーワード駆動テスト 547
 - ハンドル無効エラー 517
 - モバイル 432
- 取り消し
 - 自動保存、アクティブ データ 148

な

- 名前の変更
 - 資産 309
 - ビジュアル テスト 309
 - プロジェクト 619, 620
- 並べ替え
 - 資産 311

に

- 入力パラメータ
 - 削除 237
 - スクリプトへの追加 236
 - 定義 65
 - 編集 236
- 入力引数の型が一致しない
 - 動的呼び出し 259
- 入力フィールド
 - 検索 518

ね

- ネイティブ モバイル
 - メソッドの呼び出し 442
- ネイティブ モバイル アプリケーション

- Android、前提条件 403
- iOS、前提条件 410
- 制限事項 440
- ネイティブ再生
 - 比較、API 再生 489
- ネイティブなユーザー入力
 - 記録オプション、設定 577
 - 利点 489
- ネスト
 - キーワード 535

は

- バージョン
 - ページ 303
- ハイブリッド アプリケーション
 - Android 404
 - iOS 414
- ハイライト
 - 再生中のオブジェクト 584
- 配列パラメータ
 - 例 235
- パスワード
 - UFT Mobile、変更 426
 - 暗号化 265
- パスワードの編集
 - UFT Mobile 426
- パスワードを変更する
 - ログオン 26, 608
- パラメータ
 - Silk Central 205
 - 値、割り当て 203
 - 受け渡し、スクリプト内 229
 - キーワード間でのデータの受け渡し 534
 - キーワード駆動テスト クラス 1811
 - 組み込んだビジュアル テスト、値の割り当て 203
 - 再利用、ビジュアル テスト 207
 - 削除 205
 - 作成 202
 - 処理、キーワード 538
 - 数値の例 234
 - スクリプト、受け渡し 228
 - スクリプト、作成 228
 - スクリプト、ビジュアル テストでの使用 227
 - スクリプトでの作成と受け渡し 228
 - スクリプトとビジュアル テスト間での受け渡し 224
 - 定義 225
 - 配列の例 235
 - ビジュアル テスト間での受け渡し 205
 - ビジュアル テストで定義 206
 - ビジュアル テスト、使用 202
 - ビジュアル テスト、追加 202
 - ビジュアル テストの編集 204
 - ブール値の例 234
 - リストの例 232
 - 例 232
 - 列挙の例 233
- パラメータの受け渡し
 - スクリプト間 228
 - スクリプト内 229
- パラメータの作成

- スクリプト 228
- パラメータの使用
 - ビジュアルテスト 202
- パラメータの定義
 - データ、再利用 225
- 貼り付け
 - テストステップ 214
- 判断ロジック
 - If、Else If、Else テストステップ 159
 - 追加、テストステップ 159
 - テストステップの編集、例のシナリオ 95
 - テストステップの編集、例の詳細 96
 - ビジュアルテスト、追加 157
- 判断ロジックの編集
 - テストステップ、例 95
- ハンドル無効エラー
 - トラブルシューティング 517

ひ

- ビジュアルテスト
 - 再生 46, 276
 - 実行 276
 - 2 つめのテストの記録 54
 - Java Network Launching Protocol (JNLP) 83
 - アクティブ データ、マップ 145
 - アクティブ データ、使用 143
 - アクティブ データの更新 141, 147
 - アクティブ データの更新、式の使用 147
 - アクティブ データの更新、自動保存の回避 148
 - アクティブ データのサンプル、詳細 92
 - 新しいブラウザ ウィンドウ、起動 84
 - アプリケーションの起動 210
 - イメージ検証を追加する 366
 - 印刷 270
 - インポート 313
 - インポートとエクスポート 313
 - エラー処理、再生 268
 - エラー処理ロジック、追加 168
 - エラーのトレース 293
 - オブジェクト マッピング 344
 - 概要 90
 - 拡張 153
 - 拡張、概要 49
 - 拡張したテストの再生 52
 - 確認 44
 - 画面の更新 190
 - 画面の挿入 88
 - 画面プレビュー 643
 - キーワードとして指定 533
 - キャプチャ画面の更新、再生 190
 - キャプチャ画面の更新、テスト アプリケーション 190
 - 記録 42, 78
 - 記録、サンプル アプリケーション 43
 - 記録、出力オプション 581
 - 記録、複数のアプリケーション 82
 - 組み込み、実行 53
 - 組み込んだスクリプト、変更 208
 - 繰り返しロジック、追加 161
 - 繰り返しロジックの作成、アクティブ データ 144
 - グローバル変数 195

- グローバル変数、管理 196
- グローバル変数、作成 196
- グローバル変数、取得 197
- 結果 285, 294
- 結果、作成 291
- 結果コメント 294
- 結果への切り替え 293
- 現在の時刻の取得 185
- 現在の日付を取得 183
- 検索する 311
- 検証の追加 50
- 検証ロジック、追加 165
- 更新 49
- 更新されたアクティブ データの保存 148
- コメントの追加 153
- コントロールプロパティ データ 188
- 再生の設定 107
- 再生の停止、選択したポイント 266
- 削除 310
- 作成 78
- 作成、アセット ブラウザ 307
- 作成、複数のアプリケーション 82
- サンプル 91
- サンプル、アクティブ データ 91
- 自動エラー処理の設定、プロパティ ペイン 269
- 自動ステップ、手動で作成 186
- 手動による記録 88
- スクリプトでの使用 65, 66
- スクリプトにデータを渡す 224
- スクリプトの実行 208, 226
- スクリプト パラメータ、使用 227
- ステップ実行の制御、再生 267
- ステップのコピー 214
- ステップの削除、画面プレビュー 215
- ステップの削除、ストーリーボード 216
- ステップの削除、テスト ステップ ウィンドウ 215
- ステップの貼り付け 214
- ステップの表示 85
- ストーリーボード 645
- 相違点のマージ 217
- 操作の変更 153
- 操作を記録する 87
- 挿入 55
- 段階的実行、選択したポイントから 266
- 段階的実行、デバッグ モード 268
- データ、共有 202
- データ、再利用 191
- データの受け渡し 205
- テスト ステップ 214
- テスト ステップのアイコン 641
- テスト ステップの種類 641
- テスト プロパティ 639
- テスト ロジック デザイナを使用してエラーに自動的に対処するように設定する 269
- デバッグ 266
- デバッグ中の編集 267
- 名前の変更 309
- 並べ替え 311
- バージョン、相違点のマージ 217
- バージョン、比較 216
- パラメータ、値の割り当て 203

- パラメータ、削除 205
 - パラメータ、追加 202
 - パラメータ、定義 225
 - パラメーターの編集 204
 - パラメータの再利用 207
 - パラメータの定義 206
 - 判断ロジック、追加 157
 - 判断ロジック、例 95
 - 判断ロジック、例のシナリオ 95
 - 判断ロジック、例の詳細 96
 - 比較 216
 - 引数を eCATT スクリプトからインポートする 476
 - 引数を eCATT スクリプトにエクスポートする 477
 - 表示、複数バージョン 312
 - 開く 89, 307
 - フィルタリング 310
 - 複数のステップ、選択 214
 - プロパティ 644
 - プロパティ ペイン、表示 644
 - 変数 191
 - 保存 44, 90
 - 命名 44
 - メッセージ ボックス 209
 - モジュール式の概要 53, 72
 - 予約変数 197
 - 予約変数、戻り値型 198
 - 予約変数、リスト 197
 - 利点 90, 91
 - ローカル変数 193
 - ローカル変数、作成 51
 - ローカル変数、データの格納 52
 - ローカル変数、編集 194
 - ローケターからオブジェクト マップへの移動 83
 - ロジック 156
 - ビジュアル ナビゲータ
 - 概要 41, 637
 - 画面プレビュー 643
 - 結果 290
 - ストーリーボード 645
 - ストーリーボードでのサムネイルの移動 646
 - テスト ステップのアイコン 641
 - テスト ステップの種類 641
 - [テスト ステップ] ペイン 639
 - プロパティ 644
 - レイアウトのカスタマイズ 646
 - ビジュアル ブレークポイント
 - 検出 510
 - ビジュアル テストの拡張
 - 画面プレビューからの更新 49
 - 検証の追加 50
 - 再生 52
 - 変数、作成 51
 - ローカル変数、データの格納 52
 - ビジュアル テストの記録
 - 概要 42
 - 確認 44
 - サンプル アプリケーション 43
 - 保存 44
 - ビジュアル テストの実行
 - 値の取得、設定 282
 - 組み込み 277
 - 挿入されたスクリプト、設定 283
 - 挿入されたビジュアル テスト、設定 283
 - ビジュアル テスト パラメータ
 - 再利用する 207
 - ビジュアル テスト変数
 - ダイアログ ボックス 278
 - 日付
 - FormatDate 関数 183
 - 必要な権限
 - Silk Test の実行 595
 - Silk Test のインストール 595
 - ビデオ
 - 表示されない 432
 - 非表示
 - 入力フィールド 518
 - 表示
 - プロパティ、ビジュアル テスト ステップ 644
 - 画面イメージ、ビジュアル テスト 644
 - 結果コメント 294
 - コメント 153
 - 資産 647
 - 資産、複数バージョン 312
 - データベース エラー 615
 - データベースへの接続ユーザー 614
 - テスト ステップの説明 85
 - テスト ステップの選択 85
 - ビジュアル テスト 645
 - ビジュアル テスト、画面プレビュー 188
 - プロジェクト参照 617
 - [表示] メニュー 628
 - 開く
 - Access データベース 611
 - Oracle データベース 611
 - SQL Server データベース 610
 - キーワード 535
 - 資産 307
 - データベース 610
 - ビジュアル テスト 89
- ## ふ
- ファイアウォール
 - 競合の解決 338
 - ポート番号 340
 - ファイル
 - スクリプトからの削除 239
 - スクリプトでの名前の変更 239
 - スクリプトへの追加 238
 - [ファイル] メニュー
 - コマンド 626
 - フィルタ
 - 複製 299
 - フィルタ処理
 - 適用されたフィルタの削除 299
 - テスト ステップの表示 85
 - フィルタの適用
 - 結果 299
 - フィルタリング
 - キーワード 546
 - キーワード シーケンス、アセット ブラウザ 310
 - キーワード 駆動テスト、アセット ブラウザ 310

- 結果 297
- 結果、フィルタの作成 297
- 結果、フィルタの適用 299
- 結果、フィルタの編集 298
- 資産 310
- ブール値パラメータ
 - 例 234
- 複数のアプリケーションのテスト
 - テスト スクリプトの記録 116
- 複数のイメージ
 - 追加、イメージ資産 364
- 複数のイメージを追加する
 - イメージ資産 364
- 複数のテストの実行
 - Apple Safari 499
- 複製
 - 資産 309
- ブックマーク
 - スクリプトでのコードの移動 236
- ブラウザ
 - 起動、スクリプト 518
- ブラウザー
 - 最大化 514
 - 定義、再生 484
- ブラウザー ウィンドウ
 - サイズの指定 509
- ブラウザー テスト
 - 再生、並列 280
- ブラウザーの設定
 - 再生 484
- ブラウザ構成設定
 - xBrowser 490
- ブラウザの起動
 - 再生 518
- ブラウザの種類
 - Chrome for Android、設定 438
 - 現在の表示 514
 - 現在の表示、GetProperty 514
 - 使用法 514
- フラグ
 - 概要 154
 - 結果への挿入 155
 - 削除する 156
 - [テスト ステップ] ペインへの挿入 154
 - 表示 155
 - 表示、開始画面 155
 - 編集 155
- フラグの設定
 - プロパティ 102
- 古い資産のバージョン
 - 削除 303
- プロジェクト
 - アクティブの変更 36
 - インポート 313
 - インポートとエクスポート 313
 - 概要 35, 615
 - 管理 35, 615
 - 削除 620
 - 作成 616
 - 参照 616
 - 参照、グローバル 617

- 参照、表示 617
- 循環参照、回避 617
- 名前の変更 619, 620
- 表示 36
- ユーザー アクセス権、変更 622
- プロパティ
 - アクティブ データ 99
 - エラー時の移動先プロパティ 104
 - オブジェクト 104
 - 結果 108
 - 検証結果の説明 109
 - コマンド 101
 - 再生 105, 107
 - 再生設定の取得 103
 - 再生設定の設定 108
 - 再生の設定、挿入されたスクリプト 283
 - 再生の設定、挿入されたビジュアル テスト 283
 - 詳細設定 99
 - 全般 102
 - その他 103
 - 待機 109, 212
 - タイマー 108
 - 遅延 102
 - ビジュアル テスト 644
 - ビジュアル テストでのコントロール値 188
 - ビジュアル テストの再生 281
 - フラグの設定 102
 - 割り当て 100
- プロパティに基づく条件
 - 定義 157, 161
- [プロパティ] ペイン
 - スクリプト 650
 - ビジュアル テスト、表示 644
- プロファイル
 - Mozilla Firefox 504
- プロファイルを編集する
 - ユーザー 622
- へ
- 並列実行
 - ブラウザ 280
 - モバイル テスト 280
- 並列テスト
 - テスト済みの構成、Android 407
- ページ同期
 - xBrowser 488
- ベスト プラクティス
 - スクリプト、作成 110
- ヘッダー
 - アクティブ データ ファイル 144
- 別の場所
 - データベース 33
- ヘルプ
 - 関連資料 34
 - キーボード 36
 - トピックの印刷 76
 - 表記規則 75
 - ヘルプの表示方法 75
- [ヘルプ] メニュー 634
- 変更

- 記録の出力オプション 579
 - グループ 624
 - 再生ホットキー 591
- 変更する
 - オプション、記録の出力 581
 - オプション、詳細設定アクティブ データ 593
- 変更の表示
 - .NET スクリプト 240
 - .NET スクリプト、バージョン 240
 - オブジェクト マップ 351
 - オブジェクト マップ、バージョン 351
 - 資産 307
 - 資産、バージョン 307
 - ビジュアル テスト 216
 - ビジュアル テスト、バージョン 216
- 変更のマージ
 - .NET スクリプト 241
 - .NET スクリプト、バージョン 241
 - オブジェクト マップ 352
 - オブジェクト マップ、バージョン 352
 - ビジュアル テスト、バージョン 217
- 編集
 - アクティブ データ ファイル 141
 - アプリケーション構成 123
 - 組み込んだスクリプト 208
 - グループ 624
 - 結果フィルタ 298
 - デバッグ中のビジュアル テスト 267
 - メニュー 627
 - ユーザー プロファイル 622
 - リモート ロケーション 125
 - ローカル変数、ビジュアル テスト 194
- 変数
 - イメージ資産名、イメージ クリック 365
 - イメージ資産名、イメージ検証 367
 - キーワード間でのデータの受け渡し 534
 - キーワード駆動テストの実行 541
 - グローバル、取得 197
 - グローバル定義、管理 196
 - グローバル定義、作成 196
 - グローバル定義、取得 197
 - グローバル定義、ビジュアル テスト 195
 - コントロール プロパティ値 188
 - 再生中の監視 192
 - 削除 195
 - スクリプト 220, 221
 - スクリプトでの Public と Private 221
 - スクリプトでの使用 220, 222, 223
 - 追加 71
 - データの格納 52
 - デバッグ 57
 - ビジュアル テスト 191
 - 予約 197
 - 予約、戻り値型 198
 - 予約、リスト 197
 - ローカル、作成 51
 - ローカル、追加 193
 - ローカル定義、ビジュアル テスト 193

ほ

- ポート
 - 構成、Information Service 340
 - Open Agent 338
- 保存
 - 記録オプション 582
 - 更新データ、アクティブ データ ファイル 148
 - 再生オプション 591
 - ビジュアル テスト 90
- ホットキー
 - オプション 582
 - 設定 575

ま

- マージ
 - オブジェクト マップ 361
 - 資産、バージョン 308
- マウス移動の詳細設定
 - 設定、クロス ブラウザ テスト 490, 574
- マウス イベント遅延
 - 設定 586
- マッピング
 - アクティブ データ、ステップ テキスト 146
 - アクティブ データ、ビジュアル テスト 145

む

- 無応答のアプリケーションのタイムアウト
 - 設定 586
- 無効化
 - テクノロジー ドメイン 126
- 無視するクラス 575

め

- 命名規則
 - 資産 313
- メイン画面
 - GUI 39
- メッセージの受信
 - データベース メンテナンス 614
- メッセージ ボックス
 - アイコン、表示 99
 - 式、例 102
 - 挿入 209
 - テキスト、表示 102
- メニュー
 - ウィンドウ 634
 - 概要 626
 - カスタマイズ 633
 - 操作 628
 - 挿入 630
 - ツール 632
 - デバッグ 631
 - 表示 628
 - ファイル 626
 - ヘルプ 634
 - 編集 627
- メンテナンス

データベース、接続ユーザー 614

も

モジュール式スクリプト

エラー 74

概要 72

挿入 73

モジュール式テスト

2 つめのテストの記録 54

概要 53, 72

実行 53

挿入 55

モバイル

トラブルシューティング 432

モバイル Web

iOS 413

既存のテスト 444

既知の問題 654

クリック 443

モバイル アプリケーション

記録 422

テスト 402

モバイル テスト

Android 403

iOS 409

Web アプリ、iOS 413

Web アプリ、iOS シミュレータ 413

概要 402

再生、並列 280

接続文字列 426

デバイスの開放 430

ネイティブ アプリ、iOS シミュレータ 412

リモート ロケーション、追加 125

モバイル テスト デバイス

ネイティブ アプリ、iOS 411

モバイル デバイス

Silk Central

モバイル デバイス 432

操作する 430

定義、再生 422

に対して操作を実行する 430

モバイル ネイティブ アプリケーション

制限事項 440

モバイル ブラウザ

制限事項 439

モバイル デバイスの設定

再生 422

モバイルの記録

について 422

ゆ

ユーザー

管理 621

削除 622

追加 621

プロジェクト アクセス権、変更 622

プロファイル、編集 622

ユーザー インターフェイス

概要 31

関連トピック 625

ユーザー データ ディレクトリ

Google Chrome 501

ユーザー アカウント制御

概要 126

ユーザー アカウントを作成する

eCATT 469

ユーザーの設定

Oracle 599

ユーザーの編集

プロファイル 622

ユーザー補助

有効化 592

オブジェクト解決の向上 246

使用法 247

ユニバーサル Windows プラットフォーム

サポート 479

トラブルシューティング 480

よ

[要約] タブ

結果 287

予期しない Click 動作

Internet Explorer 517

よくある質問

オブジェクト マップ 361

クロス ブラウザ テスト 513

コードの追加、AUT 259

動的呼び出し 257

読み取りオプション

アクティブ データ、設定 140

予約変数

概要 197

戻り値 198

リスト 197

ら

ライセンス

利用可能なライセンスの種類 25

ライブラリ

アップロードする 544

ランダム数

アクティブ データ 141

り

リスト パラメーター

例 232

リテラル値の作成

条件 174

リテラル プロパティ

値の作成 174

リモート テスト

Google Chrome 492

Microsoft Edge 492

Mozilla Firefox 492

Open Agent 343

リモート ブラウザー テスト

接続文字列 492
リモート ロケーション
追加 125
編集 125
リモート エージェント
について 343

れ

例

アクティブ データ、テスト シナリオ 91
ビジュアル テスト 91
ビジュアル テスト、アクティブ データ 91
ビジュアル テスト、アクティブ データの詳細 92
ビジュアル テスト、判断ロジック 95
メッセージ ボックスの式 209

例のシナリオ

ビジュアル テスト、判断ロジック 95

例の詳細

ビジュアル テスト、判断ロジック 96

レコメンド

アルゴリズム 538

レスポンス Web デザイン

ブラウザ ウィンドウ、サイズの指定 509

ビジュアル ブレークポイント、検出 510

列挙パラメータ

例 233

列数

アクティブ データ、ビジュアル テスト 149

連絡先情報 76, 77

ろ

ローカル変数

削除 195

作成 51

追加 193

データの格納 52

ビジュアル テスト 193

ビジュアル テストの編集 194

ログ

コマンド ライン 566

設定 566

ログイン

Silk Test Workbench 26, 608

ログオン

Silk Test Workbench 26, 608

ログオン パスワード

変更する 26, 608

ロケータ

サポートされているサブセット 320

スクリプトでのサポート 320

xBrowser 517

xBrowser 内で不正 515

エイリアスの使用 573

オブジェクト タイプ 317

[オブジェクトの識別] ダイアログ ボックス 648

オブジェクト マップでの変更 350

カスタマイズする 322

基本概念 317

記録 88

検索範囲 317

構文 318

サポートしない構成子 318

サポートする構成子 318

スクリプトの構文 111

スクリプトのための記録 120

設定 575

属性の使用 318

デフォルト モードの設定 575

ビジュアル テストでのオブジェクト マップ エントリ

への移動 83

マッピング 344

ロケータ属性

Rumba コントロール 464, 524

UI オートメーション コントロール 251, 523

WPF コントロール 450, 525

記録オプション、設定 577

Silverlight コントロール 458, 522

除外される文字 134

動的 527

ロジック

エラー処理 168

概要 156

ステップのコピーおよび貼り付け 214

ロジック ツールボックス

概要 170

わ

割り当てプロパティ

概要 100