

---

# Host Access for the Cloud Web Client

3.1.1

# Table of contents

Bienvenue dans le client Web Host Access for the Cloud	4
Paramètres de connexion	5
Paramètres de connexion	5
Paramètres de connexion communs	5
Paramètres de connexion 3270 et 5250	11
Paramètres de connexion VT	13
Paramètres de connexion UTS	16
Paramètres de connexion T27	19
Paramètres de connexion ALC	20
Utilisation des sessions	22
Utilisation des touches rapides	22
Modification de l'écran	23
Déconnexion	24
Macros	25
Création de macros	25
Objets de l'API Macro	31
Exemples de macros	64
Exécuter une macro lors d'un événement	0
Paramètres d'affichage	0
Assignation de couleurs	0
Configurer les zones réactives	0
Configuration des dimensions d'écran pour les hôtes VT, UTS et T27	0
Définir les options du curseur	0
Définir les options de police	0
Définir les options du tampon de défilement arrière VT	0
Définir les options du clavier	0
Paramètres du terminal	0

Définir d'autres options d'affichage	0
Assigner des touches	0
Assigner des touches	0
Assignation de clavier de l'hôte	0
Transférer des fichiers	0
IND\$FILE	0
AS/400	0
FTP	0
Transferts par lots	0
Spécifier les options d'édition	0
Impression	0
Capturer un écran	0
Imprimer un écran	0
Impression hôte	0
Personnalisation des sessions d'hôte	0
Gestion des préférences des utilisateurs	0
Mentions légales	0

# Bienvenue dans le client Web Host Access for the Cloud

---

Le client Web Rocket® Host Access for the Cloud fournit un accès HTML5 par navigateur aux applications hôte 3270, 5250, VT, UTS, ALC et T27. Avec le produit Host Access for the Cloud, il n'est pas nécessaire de modifier le bureau, de déployer des logiciels, d'appliquer des correctifs ou d'effectuer des configurations. Vous pouvez fournir un accès utilisateur indépendant de la plate-forme à l'ensemble de vos applications hôte.

# Paramètres de connexion

---

## Paramètres de connexion

---

Des paramètres de connexion communs s'appliquent à l'ensemble des types d'hôtes.

[Paramètres de connexion communs](#)

Des paramètres spécifiques sont propres à votre type d'hôte.

[Paramètres 3270 et 5250](#)

[T27](#)

[UTS](#)

[VT](#)

[ALC](#)

## Paramètres de connexion communs

---

Les options ci-dessous sont communes à tous les types d'hôtes pris en charge.

- **Se connecter au démarrage**

Par défaut, les sessions sont configurées pour se connecter automatiquement à l'hôte lorsque vous créez ou ouvrez une session. Vous pouvez toutefois configurer une session de façon à ce qu'elle ne se connecte pas automatiquement à l'hôte. Choisissez NON pour vous connecter manuellement à l'hôte.

- **Se reconnecter lorsque l'hôte met fin à la connexion**

Lorsque cette option est définie sur Oui, Host Access for the Cloud tente de se reconnecter dès que la connexion hôte est fermée.

- **Protocole**

Dans la liste déroulante, sélectionnez le protocole à utiliser pour communiquer avec l'hôte. Pour établir une connexion hôte, le client Web et l'ordinateur hôte doivent tous deux utiliser le même protocole réseau. Les valeurs disponibles dépendent de l'hôte auquel vous vous connectez. Il s'agit des valeurs suivantes :

Protocole	Description
-----------	-------------

TN3270	TN3270 représente un type de protocole Telnet, qui est un ensemble de spécifications relatives à la communication générale entre les ordinateurs de bureau et les systèmes hôtes. Il utilise TCP/IP comme protocole de transport entre les ordinateurs de bureau et les mainframes IBM.
TN3270E	TN3270E ou Telnet étendu est destiné aux utilisateurs de TCP/IP qui se connectent à leur mainframe IBM via une passerelle Telnet conforme à la norme RFC 1647. Le protocole TN3270E vous permet de spécifier le nom du périphérique de connexion (également appelé « nom LU ») et assure la prise en charge de la clé ATTN, de la clé SYSREQ et du traitement des réponses SNA. Si vous essayez d'utiliser le protocole Telnet étendu pour vous connecter à une passerelle ne prenant pas en charge ce protocole, le protocole TN3270 standard est utilisé à la place.
TN5250	TN5250 représente un type de protocole Telnet, qui est un ensemble de spécifications relatives à la communication générale entre les ordinateurs de bureau et les systèmes hôtes. Il utilise TCP/IP comme protocole de transport entre les ordinateurs de bureau et les ordinateurs AS/400.
Shell sécurisé (VT)	<p>Vous pouvez configurer des connexions SSH si vous avez besoin de communications sécurisées et chiffrées entre un hôte VT approuvé et votre ordinateur sur un réseau non sécurisé. Les connexions SSH garantissent que l'utilisateur client et l'ordinateur hôte sont authentifiés et que toutes les données sont chiffrées. Deux options d'authentification sont disponibles :</p> <p><b>Authentification interactive via le clavier</b> : vous pouvez utiliser cette méthode d'authentification pour implémenter différents types de mécanismes d'authentification. Toute méthode d'authentification actuellement prise en charge qui ne nécessite que l'entrée de l'utilisateur peut être exécutée à l'aide de l'option d'authentification interactive via le clavier.</p> <p><b>Authentification par mot de passe</b> : cette option demande au client de fournir un mot de passe à l'hôte une fois que la connexion à celui-ci est établie. Le mot de passe est envoyé à l'hôte via le canal chiffré.</p>
Telnet (VT)	Telnet est un protocole qui fait partie de la suite TCP/IP de protocoles ouverts. En tant que protocole de flux de caractères, Telnet transmet l'entrée utilisateur des applications en mode caractères à l'hôte sur le réseau, un caractère à la fois, où elle est traitée et renvoyée en écho sur le réseau.
INT1 (UTS)	Permet d'accéder aux hôtes Unisys 1100/1200 à l'aide du protocole réseau TCP/IP.

TCPA (T27)	Utilisez ce protocole pour vous connecter aux hôtes Unisys ClearPath série NX/LX ou série A. Le processus d'authentification TCPA vérifie les informations de connexion de l'utilisateur. Lorsque ce protocole est correctement configuré, vous pouvez demander des informations d'identification de sécurité au serveur d'informations d'identification de votre application et les renvoyer au serveur. Si les informations d'identification sont valides, votre application est connectée ; vous n'avez pas besoin d'entrer d'identifiant utilisateur ou de mot de passe. Si les informations d'identification ne sont pas valides, vous devez entrer un identifiant utilisateur et un mot de passe.
MATIP (ALC)	Le protocole MATIP (Mapping of Airline Traffic over Internet Protocol) utilise TCP/IP pour la réservation aérienne, la réservation de billets et le trafic de messagerie.

#### • Sécurité TLS

Le protocole TLS permet à un client et à un serveur d'établir une connexion sécurisée et chiffrée sur un réseau public. Lorsque vous vous connectez à l'aide du protocole TLS, Host Access for the Cloud authentifie le serveur avant d'ouvrir une session, et toutes les données transmises entre Host Access for the Cloud et l'hôte sont chiffrées à l'aide du niveau de chiffrement sélectionné.

#### conseil

Lorsque la sécurité TLS est définie sur TLS 1.3 ou TLS 1.2, vous avez la possibilité de vérifier le nom d'hôte par rapport au nom du certificat de serveur. Il est vivement recommandé d'activer la vérification du nom d'hôte pour toutes les sessions.

Les options suivantes sont disponibles :

Option liée à la sécurité	Description
Aucun	Aucune connexion sécurisée n'est requise.
TLS 1.3	Connexion à l'aide de TLS 1.3. Lorsque l'option <b>Vérifier l'identité du serveur</b> est définie sur <b>Oui</b> , le client vérifie le nom du serveur ou de l'hôte par rapport au nom du certificat de serveur. Il est vivement recommandé d'activer la vérification du nom d'hôte pour toutes les sessions.

Option liée à la sécurité	Description
TLS 1.2	Connexion à l'aide de TLS 1.2. Lorsque l'option <b>Vérifier l'identité du serveur</b> est définie sur <b>Oui</b> , le client vérifie le nom du serveur ou de l'hôte par rapport au nom du certificat de serveur. Il est vivement recommandé d'activer la vérification du nom d'hôte pour toutes les sessions.

- **Activer le suivi d'émulation**

Vous pouvez choisir de générer des traces d'hôte pour une session. La valeur par défaut est Non. Sélectionnez Oui pour créer une trace d'hôte d'émulation chaque fois que la session est lancée.

## Utilisation du gestionnaire des identifiants de terminal

**MSS** Pour utiliser le gestionnaire des identifiants de terminal, vous devez avoir configuré un serveur du gestionnaire des identifiants de terminal. Reportez-vous au [Guide du gestionnaire des identifiants de terminal](#).

Le gestionnaire des identifiants de terminal fournit des identifiants aux applications client au moment de leur exécution et gère les identifiants mis en réserve pour différents types d'hôtes. Un identifiant est une donnée de connexion unique pour une session d'hôte individuelle.

Si vous décidez d'utiliser le gestionnaire des identifiants de terminal et avez configuré le serveur du gestionnaire des identifiants de terminal, vous pouvez sélectionner l'une des options ci-dessous pour configurer les critères d'acquisition d'un identifiant. Tous ces critères doivent être respectés pour garantir le renvoi d'un identifiant.

### remarque

N'oubliez pas que, en spécifiant un critère, vous indiquez que l'identifiant ne doit être attribué que si un identifiant ayant une valeur spécifique est détecté. Pour que la demande d'identifiant aboutisse, l'ensemble de critères sélectionné ici doit correspondre exactement à celui spécifié pour au moins une réserve d'identifiants dans le gestionnaire des identifiants de terminal.



## Critères du gestionnaire des identifiants de terminal

Critère	Description
Nom de la réserve	Ajoutez cet attribut et entrez le nom de la réserve pour limiter la recherche de l'identifiant à la réserve indiquée.
Adresse IP du client	L'adresse IP de l'ordinateur client est ajoutée en tant qu'élément de la demande pour un identifiant.
Adresse de l'hôte	L'adresse IP de l'hôte configuré pour cette session est ajoutée en tant qu'élément de la demande pour un identifiant.
Port de l'hôte	Le port de l'hôte configuré pour cette session est ajouté en tant qu'élément de la demande pour un identifiant.
Nom de session	Lorsque cette option est sélectionnée, cet identifiant doit être configuré pour n'être utilisé que par cette session.
Type de session	Le type de session (par exemple, IBM 3270, IBM 5250, UTS, ALS ou T27) est toujours inclus dans la demande d'identifiant.
Nom d'utilisateur	<p>Utilisez ce critère pour vérifier que seuls les identifiants créés en vue d'une utilisation exclusive par des utilisateurs spécifiques sont attribués. Le nom de l'utilisateur actuel, qui doit se trouver dans un identifiant avant de pouvoir être alloué, est le nom de l'utilisateur auquel la session est allouée au moment de l'exécution.</p> <p>Pour configurer une session en fonction de noms d'utilisateur, un nom d'utilisateur de réservation par défaut est disponible : <code>tidm-setup</code>.</p> <p>Pour que l'administrateur configure des sessions à l'aide de <code>tidm-setup</code>, des identifiants doivent être provisionnés pour <code>tidm-setup</code> dans le gestionnaire des identifiants de terminal. Vous pouvez remplacer le nom par défaut par l'un des vôtres en modifiant le fichier <code>&lt; dossier_installation &gt;/sessionserver/conf/container.properties</code> comme suit :</p> <pre>id.manager.user.name=nom_utilisateur_personnalisé, où nom_utilisateur_personnalisé est remplacé par le nom que vous souhaitez utiliser.</pre>

Nom de l'application (UTS)	Le nom de l'application hôte est utilisé en tant qu'élément de la demande pour un identifiant.
----------------------------	--

Pour déterminer le comportement de tentative de connexion si le gestionnaire des identifiants de terminal n'alloue pas d'identifiant à cette session, utilisez l'option **Si l'identifiant n'est pas alloué** :

- **Échec de la tentative de connexion** : lorsque ce paramètre est sélectionné, la session ne tente pas de se connecter si aucun identifiant n'est alloué.
- **Autoriser la tentative de connexion** : lorsque ce paramètre est sélectionné, la session tente de se connecter si aucun identifiant n'est alloué. Cette tentative peut être rejetée par l'hôte. Certains types d'hôtes permettent à un utilisateur de se connecter sans identifiant.

Pour vérifier que le gestionnaire des identifiants de terminal peut fournir un identifiant à l'aide des critères et valeurs sélectionnés, cliquez sur [Critères de test](#).

- **Envoyer des paquets KeepAlive** : utilisez ce paramètre pour assurer une vérification constante entre votre session et l'hôte de façon à repérer les problèmes de connexion lorsqu'ils surviennent. Choisissez parmi les types de paquets KeepAlive suivants :

Option	Action
Aucun	Option par défaut. Aucun paquet n'est envoyé.
Système	La pile TCP/IP effectue le suivi de la connexion hôte et envoie des paquets KeepAlive peu fréquemment. Cette option utilise moins de ressources système que les options Envoyer des paquets NOP et Envoyer des paquets de marques de synchronisation.
Envoyer des paquets NOP	Une commande NOP (No Operation) est régulièrement envoyée à l'hôte. Aucune réponse de l'hôte n'est requise, mais la pile TCP/IP peut détecter les problèmes lors de l'envoi du paquet.
Envoyer des paquets de marques de synchronisation	Une commande de marque de synchronisation est régulièrement envoyée à l'hôte afin de déterminer si la connexion est toujours active. L'hôte doit répondre à cette commande. Si aucune réponse n'est reçue ou si une erreur se produit lors de l'envoi du paquet, la connexion s'arrête.

- **Timeout KeepAlive (secondes)** : si vous choisissez d'utiliser l'option Envoyer des paquets NOP ou Envoyer des paquets de marques de synchronisation, sélectionnez l'intervalle entre les requêtes KeepAlive définies. La plage de valeurs va de 1 à 36 000 secondes (1 heure) ; la valeur par défaut est de 600 secondes.

## Test des critères du gestionnaire des identifiants de terminal

---

Le gestionnaire des identifiants de terminal fournit des identifiants aux applications client au moment de leur exécution. Utilisez cette option de test pour confirmer que le gestionnaire des identifiants de terminal peut fournir un identifiant à l'aide des critères et valeurs sélectionnés.

Les critères de la session active sont spécifiés dans le panneau Connexion après avoir sélectionné l'option **Utiliser le gestionnaire des identifiants de terminal** dans le champ Nom du périphérique (types d'hôtes 3270 et 5250), Identifiant du terminal (UTS) ou Identifiant de station (T27). Par défaut, les critères sélectionnés pour la session active sont affichés.

Cliquez sur **Test** pour confirmer que le gestionnaire des identifiants de terminal peut fournir un identifiant correspondant aux critères et valeurs configurés. Le test renvoie le nom d'un identifiant disponible correspondant aux valeurs d'attribut sélectionnées.

### Test d'autres critères et valeurs

Vous pouvez également utiliser ce panneau pour tester d'autres critères que ceux associés à la session active.

1. Sélectionnez l'un des types de sessions dans la liste Type de session, puis les critères à tester. Vous pouvez tester d'autres valeurs que vous souhaitez utiliser dans un exemple de requête du gestionnaire des identifiants de terminal.
2. Cliquez sur **Test** pour confirmer que le gestionnaire des identifiants de terminal peut fournir un identifiant correspondant aux critères et valeurs sélectionnés. Le test renvoie le nom d'un identifiant disponible correspondant aux valeurs sélectionnées.

## Paramètres de connexion 3270 et 5250

---

Outre les [paramètres de connexion communs](#), les types d'hôtes 3270 et 5250 exigent ces paramètres spécifiques.

- **Modèle de terminal**

Spécifiez le modèle de terminal (également appelé « station d'affichage ») que Host Access for the Cloud doit émuler. Différents modèles de terminal sont disponibles en fonction du type d'hôte.

Si vous choisissez **Modèle personnalisé**, vous pouvez définir le nombre de colonnes et de lignes pour personnaliser le modèle de terminal.

- **Utiliser l'ouverture de session automatique Kerberos (5250 uniquement)** 

Si ce paramètre a la valeur **Oui**, l'utilisateur n'a pas besoin d'entrer des informations d'identification d'ouverture de session. L'ouverture de session automatique Kerberos est

configurée dans la Console d'administration MSS > Host Access for the Cloud. Si vous configurez HACloud pour utiliser le protocole d'authentification Kerberos, vous devez comprendre certains termes et respecter des conditions préalables avant de configurer cette option. Ces options sont détaillées dans la documentation du panneau Console d'administration MSS > Host Access for the Cloud, accessible à partir du bouton Aide. Pour plus d'informations, reportez-vous au Guide de déploiement.

- **Identifiant du terminal** (3270 uniquement)

Lorsque Host Access for the Cloud se connecte à un hôte Telnet, le protocole Telnet et l'hôte définissent un identifiant de terminal à utiliser lors de la connexion Telnet initiale. En général, cette définition aboutit à l'utilisation de l'identifiant de terminal approprié. Vous devriez donc laisser cette zone de texte vide.

- **Sécurité TLS**

Le protocole TLS permet à un client et à un serveur d'établir une connexion sécurisée et chiffrée sur un réseau public. Lorsque vous vous connectez à l'aide du protocole TLS, Host Access for the Cloud authentifie le serveur avant d'ouvrir une session, et toutes les données transmises entre Host Access for the Cloud et l'hôte sont chiffrées à l'aide du niveau de chiffrement sélectionné. Pour plus d'informations sur ce paramètre commun, reportez-vous à la rubrique [Paramètres de connexion communs](#).

- **Nom du périphérique**

Si vous avez sélectionné TN3270, TN3270E ou TN5250 comme protocole, indiquez le nom du périphérique à utiliser lorsque la session se connecte à l'hôte. Le nom du périphérique est également appelé LU de l'hôte ou réserve. Vous pouvez également choisir l'une des options suivantes :

- **Générer un nom de périphérique unique** : génère automatiquement un nom de périphérique unique.
- **Utiliser le gestionnaire des identifiants de terminal** : affiche des paramètres supplémentaires à définir. Reportez-vous à la section [Utilisation du gestionnaire des identifiants de terminal](#)
- **Toujours demander l'identifiant à l'utilisateur** : l'utilisateur final est invité à entrer l'identifiant de périphérique à chaque fois qu'une connexion est établie.
- **Si l'identifiant n'est pas indiqué, le demander à l'utilisateur** : l'utilisateur final est invité à entrer l'identifiant lors de la première tentative de connexion, après quoi la valeur est enregistrée. La valeur enregistrée continue à être utilisée sans invite supplémentaire.

Si vous ne spécifiez pas de nom de périphérique pour la session, l'hôte en assigne un dynamiquement à la session. Un nom de périphérique défini dans une macro remplace ce paramètre.

## Paramètres de connexion VT

Outre les [paramètres de connexion communs](#), les hôtes VT exigent des paramètres supplémentaires. Ces paramètres varient en fonction du protocole que vous utilisez : Telnet ou SSH. Sauf mention contraire, les paramètres s'appliquent aux deux protocoles.

### Options de configuration de session VT

Paramètre VT	Description
Identifiant du terminal	Ce paramètre détermine la réponse que Host Access for the Cloud envoie à l'hôte après une demande d'attributs de périphérique primaire. La réponse permet à l'hôte de savoir quelles fonctions de terminal il peut exécuter. La réponse de Host Access for the Cloud pour chaque identifiant de terminal est identique à celle du terminal VT. Certaines applications exigent une réponse donnée à une demande d'attributs de périphérique. Le paramètre Identifiant du terminal est indépendant du paramètre Type de terminal. Les options sont les suivantes : VT220, VT420, VT100, DEC-VT100 et VT52.
Autoriser les hôtes inconnus (SSH)	Ce paramètre permet à l'administrateur de déterminer si le client Web autorise les hôtes inconnus. Les options sont les suivantes : <b>Oui</b> : les hôtes inconnus et toutes les connexions SSH sont autorisés. Les utilisateurs du client Web ne sont pas invités à indiquer si les hôtes doivent être approuvés. <b>Demander</b> : l'utilisateur du client Web est invité à indiquer si l'hôte doit être approuvé lorsqu'il se connecte pour la première fois à un hôte inconnu. S'il choisit d'approuver l'hôte, sa clé publique est stockée dans ses préférences utilisateur, et les connexions ultérieures ne déclenchent pas d'invite, sauf si la clé d'hôte change. <b>Non</b> : aucun hôte inconnu n'est autorisé. Seuls les hôtes que l'administrateur choisit d'approuver lors de la configuration de la session sont autorisés. Les utilisateurs finals ne sont jamais invités à interagir, et la session se connecte ou non en fonction des choix de l'administrateur.
Supprimer les messages de bannière (SSH)	Lorsque cette option est activée, la bannière SSH ne s'affiche pas. Cette option est très utile pour enregistrer des macros de connexion SSH.

<p>Commande Echo locale (Telnet)</p>	<p>Automatique (par défaut). Ce paramètre indique comment Host Access for the Cloud répond à une commande Echo distante provenant d'un hôte Telnet. L'option Automatique tente de négocier la commande Echo distante, mais effectue ce que l'hôte ordonne. L'option Oui signifie que Host Access for the Cloud négocie la commande Echo locale avec l'hôte, mais qu'il effectue toujours l'écho. Enfin, avec l'option Non, Host Access for the Cloud négocie la commande Echo distante avec l'hôte, mais n'effectue pas l'écho.</p>
<p>Renégocier la commande Echo (Telnet)</p>	<p>Non (par défaut). Lorsque l'option Oui est définie, les mots de passe ne s'affichent pas sur l'écran local, mais le reste du texte entré est visible. Host Access for the Cloud prend en charge l'option Telnet de suppression de l'écho local (SLE) lorsque vous êtes connecté à un hôte en mode semi-duplex. Cela signifie que Host Access for the Cloud supprime l'écho de caractères sur l'ordinateur hôte et qu'avec la prise en charge de SLE, Host Access for the Cloud peut être invité à supprimer l'écho localement.</p>
<p>Définir la taille de la fenêtre de l'hôte</p>	<p>Oui (par défaut). Ce paramètre transmet le nombre de lignes et de colonnes à l'hôte Telnet lorsqu'ils changent. Cela permet à l'hôte Telnet de contrôler correctement le curseur si la taille de la fenêtre est modifiée.</p>
<p>Algorithmes de touches sur l'hôte (SSH)</p>	<p>Ce paramètre détermine les algorithmes de clé d'hôte à utiliser lors de la connexion à un serveur SSH. Les options sont les suivantes :</p> <ul style="list-style-type: none"> <li>EC, RSA, DSS (par défaut) : il s'agit de l'option la plus sécurisée. Utilisez-la lors d'une mise à niveau pour plus de sécurité.</li> <li>RSA, DSS : cette option permet de continuer à utiliser une clé existante pour les nouvelles sessions SSH.</li> </ul>
<p>Requête fichier binaire (Telnet)</p>	<p>Non (par défaut). Telnet définit un chemin de données 7 bits entre l'hôte et le terminal. Ce type de chemin de données n'est pas compatible avec certains jeux de caractères nationaux. Heureusement, de nombreux hôtes acceptent les données 8 bits sans mettre à zéro le 8e bit, ce qui résout le problème. Cependant, dans certains cas, il est nécessaire d'obliger l'hôte à utiliser un chemin de données 8 bits en sélectionnant cette case.</p>
<p>Envoyer un saut de ligne après un retour chariot (Telnet)</p>	<p>Non (par défaut). Un hôte Telnet « authentique » attend une séquence de caractères CrNu (retour chariot/caractère nul) pour indiquer la fin d'une ligne transmise par un terminal. Certains hôtes sur Internet, qui ne sont pas des hôtes Telnet « authentiques », attendent une séquence de caractères Lf (saut de ligne) après Cr à la fin d'une ligne. Si vous vous connectez à ce type d'hôte Telnet, sélectionnez l'option Oui.</p>

Ctrl-Attn envoie (Telnet)	Choisissez la séquence envoyée par Ctrl-Attn à l'hôte lors de son activation. Les options sont les suivantes : Rupture Telnet (par défaut), Interrompre le processus ou Rien.
Jeu de caractères de l'hôte	La valeur par défaut du jeu de caractères de l'hôte dépend du type de terminal que vous émulez. Ce paramètre reflète l'état actuel du terminal du jeu de caractères de l'hôte VT, qui peut être modifié par l'hôte. Le paramètre par défaut associé, enregistré avec le modèle, est DÉC supplément.
Réponse automatique	Non (par défaut). Ce paramètre détermine si le message de réponse (défini à l'aide de la propriété Réponse) est automatiquement transmis à l'hôte après l'établissement d'une communication.
Chaîne de réponse	<p>Ce paramètre vous permet d'entrer un message de réponse si l'hôte attend une réponse après avoir transmis un caractère ENQ.</p> <p>La chaîne de réponse prend en charge les caractères dont les codes sont inférieurs ou égaux à 0xFFFF par le biais de séquences d'échappement Unicode. La séquence d'échappement commence par \u suivi d'exactly quatre chiffres hexadécimaux. Vous pouvez incorporer des séquences d'échappement Unicode dans n'importe quelle chaîne. Par exemple, la séquence \u0045 incorporée est interprétée comme un caractère E incorporé, car 45 est le code hexadécimal du caractère E.</p> <p>Pour transmettre des séquences d'échappement Unicode à l'hôte, échappez la séquence avec une barre oblique inverse. Par exemple, pour envoyer le littéral de chaîne \u001C à l'hôte, assignez une touche à \ \u001C. Host Access for the Cloud convertit ce littéral en chaîne \u001C lorsque vous appuyez sur cette touche et envoie les 6 caractères de la chaîne résultante à l'hôte.</p>

## Informations supplémentaires

### Descriptions TLS

## Paramètres de connexion UTS

Outre les [paramètres de connexion communs](#), les hôtes UTS exigent les paramètres supplémentaires ci-dessous :

### Options de configuration de session UTS INT1

Option UTS INT1	Description
Application	<p>Nom de l'application hôte ou du mode de fonctionnement hôte auquel accéder.</p> <p>Il s'agit du mot ou de l'expression que l'ordinateur local envoie à l'hôte lorsque vous établissez pour la première fois la communication avec l'hôte. Si vous utilisez un terminal hôte, il s'agit du nom \$\$OPEN de l'application. Le nom de l'application est généralement le même que celui de l'environnement. Cependant, ils peuvent être différents. Par exemple, le nom de l'environnement peut être MAPPER et celui de l'application UDSSRC. Lors d'une session d'émulation de terminal, vous saisissez \$\$OPEN MAPPER lors de l'affichage de l'invite, et INT1 envoie UDSSRC à l'hôte une fois la connexion établie.</p>
TSAP	<p>Point d'accès au service de transport (TSAP) souhaité, jusqu'à 32 caractères (comme TIPCSU pour les connexions TIP, RSDCSU pour les connexions Demand). Un TSAP n'est requis que si vous vous connectez à un contrôleur LAN hôte (HLC) ou à un processeur de communications distribuées (DCP) en mode Routeur IP. Si vous n'êtes pas sûr de la valeur à utiliser, contactez votre administrateur hôte.</p>
Transaction initiale	<p>Caractère, mot ou expression que l'ordinateur local envoie à l'hôte lors de la première communication avec l'hôte (maximum 15 caractères). Ce paramètre est facultatif et est principalement utilisé avec TIP. Par exemple, vous pouvez entrer ^ pour lancer MAPPER. Ce paramètre peut également être utilisé pour transmettre des mots de passe.</p>



Démarrer la transaction	<p>Lorsque vous configurez une transaction initiale, par défaut, les données sont envoyées dès que la connexion à la session est établie. Vous pouvez déterminer le moment auquel envoyer une transaction initiale à l'aide d'une chaîne particulière pour déclencher la transaction initiale.</p> <p>Par exemple, pour attendre une connexion réussie avant d'envoyer les données de transaction initiale, saisissez la chaîne à utiliser pour identifier une connexion réussie.</p> <p>Vous pouvez utiliser ce paramètre avec l'option <b>Envoyer une transaction initiale</b>.</p>
Envoyer une transaction initiale	<p>Vous pouvez déterminer le moment auquel envoyer la transaction initiale :</p> <p><b>Immédiatement</b> : option par défaut.</p> <p><b>À la réception du caractère de début de saisie</b> : cette option est utile lorsque les transactions multiligne doivent être terminées avant l'envoi de la chaîne.</p> <p><b>Après le nombre de millisecondes spécifié</b></p>

**Identifiant  
du terminal**

Choisissez des options pour spécifier un identifiant de terminal ou pour utiliser le gestionnaire des identifiants de terminal. Pour spécifier un identifiant de terminal, tapez-le dans le champ **Spécifier l'identifiant du terminal**.

**Spécifier l'identifiant du terminal**

Identifiant du terminal (généralement 8 caractères alphanumériques maximum) à utiliser pour la session de communication associée à ce chemin. Connu également sous le nom de TID ou PID, chaque identifiant de terminal doit être unique à l'hôte.

**Si l'identifiant n'est pas indiqué, le demander à l'utilisateur**

Lorsque vous sélectionnez cette option, l'utilisateur final est invité à entrer l'identifiant lors de la première tentative de connexion, après quoi la valeur est enregistrée. La valeur enregistrée continue à être utilisée sans invite supplémentaire.

**Toujours demander l'identifiant à l'utilisateur**

Lorsque vous sélectionnez cette option, l'utilisateur final est invité à entrer l'identifiant de terminal à chaque fois qu'une connexion est établie.

**Utiliser le gestionnaire des identifiants de terminal**

Si vous choisissez cette option, vous devez sélectionner les attributs d'identifiant de terminal à utiliser pour obtenir un identifiant. Voir Attributs du gestionnaire des identifiants de terminal.

Pour tester les attributs, cliquez sur **Test**.

**Informations supplémentaires**

[Attributs du gestionnaire des identifiants de terminal](#)

[Descriptions TLS](#)

## Paramètres de connexion T27

---

Outre les [paramètres de connexion communs](#), vous pouvez configurer les options de connexion T27 supplémentaires ci-dessous :

### Paramètres de connexion T27

Option T27	Description
Type de terminal	Sélectionnez le type de terminal à émuler pendant la session. L'émulation T27 prend en charge les types de terminaux Unisys TD830, TD830 ASCII, TD830 INTL et TD830 NDL.
Requête fichier binaire	Vous devez activer l'option Requête fichier binaire lorsque vous avez besoin d'une impression directe. La valeur par défaut est Non. TCPA définit un chemin de données 7 bits entre l'hôte et l'émulateur de terminal. Ce type de chemin de données n'est pas compatible avec certains jeux de caractères nationaux. Toutefois, de nombreux hôtes acceptent les données 8 bits sans mettre à zéro le 8e bit, ce qui résout le problème. Cependant, il peut être nécessaire de forcer l'hôte à utiliser un chemin de données 8 bits ; pour ce faire, sélectionnez cette option.
Largeur de ligne	Sélectionnez le nombre de caractères que l'hôte envoie au client. La valeur par défaut est de 80 caractères.
Sécurité TLS	Pour obtenir une description des différentes options, reportez-vous à la rubrique <b>Descriptions TLS</b> .

**Identifiant de station**

Choisissez une option pour spécifier un identifiant de station ou utilisez le gestionnaire des identifiants de terminal. Pour spécifier un identifiant de station, choisissez **Spécifier l'identifiant de la station**, puis tapez le nom correspondant dans le champ Identifiant de station.

Chaque identifiant de station doit être unique pour l'hôte et se compose généralement d'un maximum de huit caractères alphanumériques.

Si l'identifiant n'est pas indiqué, le demander à l'utilisateur

Lorsque vous sélectionnez cette option, l'utilisateur final est invité à entrer l'identifiant lors de la première tentative de connexion, après quoi la valeur est enregistrée. La valeur enregistrée continue à être utilisée sans invite supplémentaire.

Toujours demander l'identifiant à l'utilisateur

Lorsque vous sélectionnez cette option, l'utilisateur final est invité à entrer l'identifiant de station à chaque fois qu'une connexion est établie.

Utiliser le gestionnaire des identifiants de terminal

Lorsque vous sélectionnez cette option, vous devez configurer plusieurs critères d'identifiant de terminal. Pour obtenir la description des différentes options, reportez-vous à la section **Critères du gestionnaire des identifiants de terminal**.

Si vous ne spécifiez pas d'identifiant de station pour la session, l'hôte en assigne un dynamiquement à la session.

**Informations supplémentaires**

- [Descriptions TLS](#)
- [Critères du gestionnaire des identifiants de terminal](#)

## Paramètres de connexion ALC

---

Outre les [paramètres de connexion communs](#), les hôtes ALC exigent les paramètres supplémentaires ci-dessous :

## Paramètres de connexion ALC

Option ALC	Description
Sécurité TLS	Pour obtenir une description des différentes options, reportez-vous à la rubrique <b>Descriptions TLS</b> .
Codage des caractères	Choisissez ASCII, EBCDIC ou IPARS (valeur par défaut) comme jeu de codes.
Fichier de configuration	Entrez le fichier de configuration (CNF) qui associe les informations de configuration appropriées à un type d'hôte spécifique.
Adresse du terminal	<p>Indiquez si vous souhaitez spécifier l'adresse du terminal ou utiliser le gestionnaire des identifiants de terminal.</p> <p><b>Adresse du terminal</b> : indiquez si vous utilisez le mode d'adressage sur 2 ou 4 octets.</p> <p>Bien qu'une adresse unique de 5 octets soit requise lorsque vous spécifiez l'identifiant de terminal au lieu d'utiliser le gestionnaire des identifiants, cette option spécifie le nombre d'octets de l'adresse d'identifiant de terminal de 5 octets envoyés avec chaque message à des fins de multiplexage. Si vous spécifiez le mode d'adressage sur 2 octets, seuls les 2 derniers octets de l'adresse de grappe ASCU (Agent Set Control Unit) (A1, A2) sont envoyés. Si vous spécifiez le mode d'adressage sur 4 octets, l'adresse de grappe ASCU complète (H1, H2, A1, A2) est envoyée. Spécifiez l'adresse de terminal unique de 5 octets pour cette session. L'adresse du terminal est composée de cinq valeurs à 2 chiffres hexadécimaux : H1, H2, A1, A2 et TA (adresse du terminal). Cette adresse unique est généralement assignée par l'administrateur réseau.</p> <p><b>Gestionnaire des identifiants de terminal</b> : fournit des ID aux applications client au moment de leur exécution. Si vous choisissez cette option, vous devez définir d'autres options de configuration. Pour obtenir la description de ces options, reportez-vous à la section <b>Critères du gestionnaire des identifiants de terminal</b>.</p>

### Informations supplémentaires

- [Descriptions TLS](#)
- [Critères du gestionnaire des identifiants de terminal](#)

# Utilisation des sessions

---

Toutes les sessions auxquelles vous avez accès sont disponibles dans la liste Sessions disponibles. Les sessions sont initialement créées et configurées par votre administrateur système et accessibles via une URL distribuée (par exemple, `https://<DNS_grappe>/webclient`).

## Pour ouvrir une session

1. Sélectionnez la session souhaitée, puis cliquez dessus pour l'ouvrir.
2. Interagissez avec votre application hôte à l'aide de la session ouverte.
3. Vous pouvez créer plusieurs instances d'une session configurée.

Vous pouvez ouvrir plusieurs sessions simultanément et passer facilement de l'une à l'autre à l'aide des onglets organisés dans la partie supérieure de l'écran. La session actuelle correspond toujours à l'onglet le plus à gauche et est indiquée par un fond blanc et un texte en gras. Chaque session reste active pendant 30 minutes.


Utilisez la barre d'outils pour accéder aux différentes options disponibles lorsque vous interagissez avec la session. Vous pouvez vous déconnecter d'une session, fermer la session, activer les touches rapides et accéder à d'autres paramètres. Certaines options ne sont disponibles que si votre administrateur vous a autorisé à y accéder.

## Utilisation des touches rapides

---

Le clavier de terminal à touches rapides fournit une représentation graphique des touches d'un clavier d'hôte et vous permet d'accéder rapidement aux touches de terminal.

Cliquez sur une touche de terminal du clavier de touches rapides pour envoyer la touche correspondante à l'hôte. Les info-bulles, disponibles en passant le pointeur sur une touche, fournissent une description de l'assignation.

Les touches rapides sont disponibles pour chaque type d'hôte pris en charge. Elles sont accessibles en cliquant sur l'icône  de la barre d'outils.

# Modification de l'écran

## 💡 remarque

Chaque navigateur gère les fonctions Copier, Coller et Couper différemment et, dans certains cas, ne prend pas en charge l'utilisation des boutons de barre d'outils ou du menu contextuel. Il est vivement recommandé d'utiliser des raccourcis clavier pour ces fonctions. Les raccourcis clavier varient en fonction du système d'exploitation que vous utilisez. Sous Windows, il s'agit des commandes suivantes : CTRL + C pour copier, CTRL + V pour coller et CTRL + X pour couper.

Il est plus courant de rencontrer des problèmes avec la fonction Coller qu'avec les fonctions Couper et Copier. Si le bouton Coller de la barre d'outils n'est pas visible, il est probable que la sécurité du navigateur empêche l'accès en lecture au Presse-papiers système. Certains navigateurs se comportent différemment lorsqu'ils doivent fournir l'accès au Presse-papiers. Cependant, la fonction Coller est presque toujours disponible à l'aide des raccourcis clavier (Contrôle + V sous Windows et Commande + V sous Mac). Cela suppose que vous n'avez pas réassigné ces touches. Vous pouvez également utiliser l'élément de menu ou le bouton Coller intégré du navigateur.

### Pour copier une zone du terminal

1. Mettez en surbrillance la zone de l'écran du terminal que vous souhaitez copier.
2. Cliquez sur **Copier** dans la barre d'outils ou sélectionnez **Copier** dans le menu contextuel accessible dans l'écran du terminal. Vous pouvez également utiliser le raccourci clavier **CTRL + C**.

### Pour coller du contenu dans l'écran du terminal

1. Positionnez le curseur à l'endroit où vous souhaitez coller du contenu.
2. Si le navigateur prend en charge la fonction Coller, cliquez sur **Coller** dans la barre d'outils ou sélectionnez **Coller** dans le menu contextuel accessible dans l'écran du terminal. Si le navigateur ne prend pas en charge cette fonction, ces options ne sont pas disponibles et vous devez utiliser le raccourci **CTRL + V**.

### Pour couper des zones de l'écran du terminal

## 💡 remarque

Cette fonction est disponible pour tous les types de terminaux pris en charge, à l'exception des hôtes VT.

1. Mettez en surbrillance la zone de l'écran du terminal que vous souhaitez couper.

2. Cliquez sur **Couper** dans la barre d'outils ou sélectionnez **Couper** dans le menu contextuel accessible dans l'écran du terminal. Vous pouvez également utiliser le raccourci clavier **CTRL + X**.

### Informations supplémentaires

[Spécifier les options d'édition](#)

## Déconnexion

---

Dans le coin supérieur droit de l'écran, ouvrez la liste déroulante associée à votre nom d'utilisateur, puis sélectionnez **Déconnexion** pour arrêter d'utiliser l'application hôte.



# Macros

---

## Création de macros

---

Une macro est une série d'opérations du clavier que vous enregistrez puis exécutez. Vous pouvez utiliser ces programmes de macro JavaScript pour automatiser les interactions utilisateur avec le terminal. Vous pouvez accéder aux macros et les exécuter à partir de tous les périphériques pris en charge.

Host Access for the Cloud enregistre et sauvegarde les macros avancées au format JavaScript, ce qui facilite leur modification et enrichit les macros enregistrées. Vous pouvez enregistrer des macros pour les lire ultérieurement, ainsi qu'exécuter des macros au démarrage ou lorsque la session se connecte ou se déconnecte de l'hôte. Vous pouvez également écrire des macros de toutes pièces pour effectuer des tâches complexes que l'enregistreur ne peut pas capturer.

Les macros sont accessibles aux utilisateurs de deux manières : créées par un administrateur ou enregistrées par les utilisateurs pour leur propre usage privé. Toutes les macros avancées sont associées à une session et poursuivent le même objectif, à savoir automatiser les interactions avec l'hôte. La seule différence entre les deux variantes concerne la personne qui peut y accéder et qui gère leur création et leur disponibilité :

- **Macros créées par les administrateurs**

Les administrateurs créent des macros lorsqu'ils créent la session. Elles sont spécifiques d'une session et sont accessibles à tous les utilisateurs qui ont accès à la session à partir de l'icône Macro de la barre d'outils. Les administrateurs peuvent désigner des macros à exécuter au démarrage ou lorsque la session se connecte ou se déconnecte de l'hôte.

- **Macros créées par les utilisateurs**

Les macros utilisateur final sont créées par des utilisateurs pour les sessions auxquelles ils sont autorisés à accéder. L'administrateur accorde l'autorisation de créer des macros en définissant une règle de préférence utilisateur. Les utilisateurs peuvent accéder à la session à l'aide de leurs informations d'identification ou du rôle Invité. Les macros que les utilisateurs invités créent sont disponibles pour tous les utilisateurs invités. Les utilisateurs connectés à l'aide de leurs informations d'identification ne peuvent voir que les macros qu'ils ont créées.

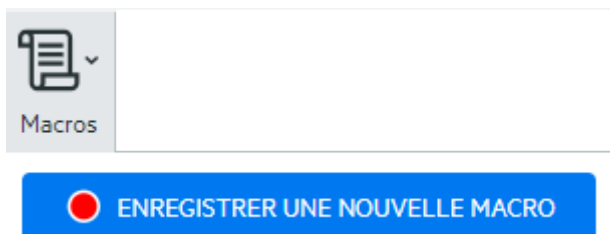
Les macros avancées sont répertoriées dans l'ordre alphabétique dans la liste déroulante disponible dans la barre d'outils. Les macros créées par l'utilisateur final sont répertoriées en premier, suivies d'un indicateur de trois points gris verticaux qui permet d'afficher les options Modifier et Supprimer. Les macros créées par l'administrateur sont répertoriées sans indicateur, car elles ne peuvent pas être modifiées par l'utilisateur final.


## Utilisation des macros

Suivez les procédures ci-dessous pour enregistrer, modifier et exécuter des macros.

### Enregistrer

1. Cliquez sur l'icône Macro de la barre d'outils, puis sur Enregistrer une nouvelle macro.




2. Parcourez l'application hôte pour enregistrer la série d'étapes à inclure dans la macro.
3. Cliquez sur  dans la barre d'outils pour arrêter l'enregistrement. Le point rouge clignote pour indiquer que l'enregistrement est en cours.
4. À l'invite, tapez le nom de la macro.

### Modifier

1. Dans la liste déroulante Macro, sélectionnez la macro à modifier.




2. Cliquez sur les trois points verticaux pour développer le champ.
3. Cliquez sur  **Modifier** pour ouvrir l'éditeur de macros (dans le panneau de gauche).
4. Utilisez JavaScript pour apporter les modifications nécessaires. Vous pouvez exécuter et enregistrer la macro modifiée à l'aide des icônes de barre d'outils dans le volet supérieur de l'éditeur.

### Exécuter

Pour exécuter une macro, choisissez-la dans la liste déroulante, puis cliquez sur .

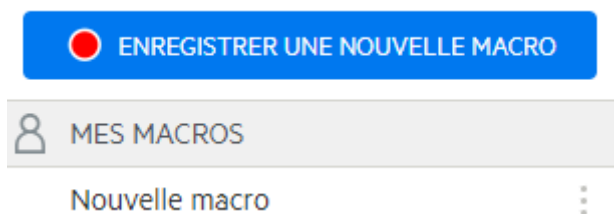
Vous pouvez également assigner des touches qui déclenchent automatiquement une macro déjà enregistrée. Dans la boîte de dialogue des paramètres d'assignation de touches, choisissez **Exécuter la macro** dans la liste déroulante **Opération**. Choisissez une macro à associer à l'assignation de touches dans la liste **Valeur**.

## Arrêter

Vous pouvez arrêter une macro avant qu'elle ne se termine à partir de l'éditeur de macros ou de la barre d'outils. Cliquez sur  pour arrêter la macro. Pour réexécuter la macro, revenez à l'écran de démarrage de la macro.

## Supprimer

1. Dans la liste déroulante Macro, sélectionnez la macro à supprimer.
2. Développez le champ en cliquant sur l'icône à trois points verticaux.



3. Cliquez sur **Supprimer**.

## Afficher

La liste déroulante Macro est accessible à partir de la barre d'outils à tous les utilisateurs autorisés à enregistrer des macros ou qui accèdent à une session dans laquelle des macros ont été pré-enregistrées par l'administrateur afin de les utiliser avec cette session.

Les macros sont répertoriées sous **MES MACROS** ou **MACROS** en fonction de la façon dont elles ont été enregistrées.

Tous les utilisateurs, qu'ils soient connectés à l'aide de leurs informations d'identification ou en tant qu'invités, peuvent voir les macros associées à la session. Les macros qui figurent sous l'en-tête **MES MACROS** sont répertoriées dans l'ordre alphabétique par nom et sont visibles par les utilisateurs qui les ont enregistrées. Les macros enregistrées par l'administrateur et jointes à une session sont répertoriées dans l'ordre alphabétique sous **MACROS**.

## Débogage des macros

Étant donné que les macros sont écrites en JavaScript et exécutées dans le navigateur, la meilleure façon de les déboguer et de les dépanner consiste à utiliser les outils intégrés de votre navigateur Web. Les navigateurs modernes intègrent un ensemble d'outils très performants pour le débogage du code JavaScript. Vous pouvez placer des points d'arrêt, parcourir le code et récupérer des informations de débogage.

### conseil

JavaScript établit la distinction entre majuscules et minuscules. Tenez compte de ce point lorsque vous modifiez du code JavaScript.

### Pour déboguer une macro :

1. Ouvrez la macro pour modification. Pour obtenir les instructions correspondantes, reportez-vous à la section [Utilisation des macros](#).
2. Ouvrez les outils de développement de votre navigateur.

Navigateur	Procédure
Mozilla Firefox	Dans la barre d'outils, ouvrez le menu, puis choisissez Outils de développement.  Dans le menu des outils de développement Web, choisissez Débugueur. Le débogueur s'ouvre dans un panneau inférieur.
Google Chrome	Dans la barre d'outils, ouvrez le menu, puis choisissez Plus d'outils.  Choisissez Outils de développement pour ouvrir le débogueur.

1. Utilisez l'un de ces outils dans votre code de macro, puis exécutez le code.

- debugger

L'approche de débogage la plus approfondie consiste à utiliser l'instruction `« debugger; »`. Lorsque vous insérez ces instructions dans votre code de macro, puis que vous l'exécutez, les outils de développement du navigateur s'ouvrent et l'exécution s'arrête sur ces lignes. Vous pouvez parcourir la macro, afficher la valeur des variables locales et tout autre élément à vérifier.

Nous vous encourageons à placer plusieurs instructions `debugger;` dans votre code pour vous aider à trouver la ligne correcte. La nature asynchrone de JavaScript peut compliquer la navigation dans le code. Pour contourner ce problème, utilisez plusieurs instructions `debugger;` placées avec soin.

**Exemple 1 :** `debugger`

```
var hostCommand = menuSelection + '[enter]';
debugger; // <- Le débogueur du navigateur s'arrête ici
ps.sendKeys(hostCommand);
```

- `console.log()`, `alert()`

Ces deux fonctions sont généralement utilisées pour déboguer du code JavaScript. Bien qu'elles ne soient pas aussi flexibles que l'instruction `debugger`, elles fournissent un moyen rapide de récupérer des informations de débogage. Ces fonctions affichent les informations dans l'onglet « Console » JavaScript des outils de développement du navigateur.

**Exemple 2 :** `console.log()`, `alert()`

```
var hostCommand = menuSelection + '[enter]';
console.log('Commande :' + hostCommand); // <- Génère la sortie de la
chaîne dans l'onglet "Console".
alert('Commande:' +
hostCommand); // Affiche une petite fenêtre contenant les données.
ps.sendKeys(hostCommand);
```

- `ui.message()`

L'API Macro de Host Access for the Cloud fournit une fonction `ui.message()` très similaire à la fonction `alert()` de JavaScript. Vous pouvez également utiliser `ui.message()` pour récupérer les informations de débogage.

**Exemple 3 :** `ui.message()`

```
var hostCommand = menuSelection + '[enter]';
ui.message('Commande :' + hostCommand); // <- Affiche une fenêtre de
message.
ps.sendKeys(hostCommand);
```

### Point à retenir :

- Navigation et mots-clés « yield »

Bien qu'elles simplifient la compréhension des macros, les instructions `yield` peuvent compliquer la navigation dans le code avec le débogueur. Envisagez d'utiliser plusieurs instructions `debugger` ou des instructions `debugger` placées avec soin d'appels `console.log()` pour récupérer les informations de débogage appropriées.

# Utilisation de l'API Macro

Dans Host Access for the Cloud, les macros sont enregistrées et écrites à l'aide de JavaScript.

L'API Macro se compose d'un ensemble d'objets que vous pouvez utiliser pour interagir avec l'hôte, attendre l'état de l'écran et interagir avec l'utilisateur.

## À propos des objets Promise et des mots-clés yield

Étant donné que JavaScript est un thread unique et utilise des « fonctions de rappel » et des « objets Promise » pour gérer le flux d'exécution dans le code, il peut souvent être difficile de suivre le code. Host Access for the Cloud associe le concept d'objets « Promise » au mot-clé « yield » de façon à pouvoir organiser le code de macro de façon plus linéaire.

- **Objets Promise**

Les objets Promise sont des modèles visant à simplifier les fonctions qui renvoient leurs résultats de façon asynchrone à un moment donné dans le futur. Toutes les fonctions « wait » et « ui » de l'API Macro renvoient des objets Promise.

- **Mot-clé yield**

Les macros utilisent le mot-clé yield pour bloquer l'exécution de la macro jusqu'à la résolution ou la fin d'un objet Promise. Ainsi, le placement du mot-clé yield devant une fonction « wait » ou « ui » suspend l'exécution de la macro jusqu'à la fin de l'exécution de cette fonction. Vous pouvez placer le mot-clé yield devant n'importe quelle fonction qui renvoie un objet Promise, même vos propres fonctions personnalisées.

### remarque

La fonction `createMacro()` permet de bloquer l'exécution d'une macro en associant le mot-clé yield à des objets Promise.

## Erreurs

Les erreurs sont traitées dans les macros à l'aide d'une instruction try/catch. Certaines fonctions de l'API peuvent générer des erreurs si, par exemple, les conditions correspondantes ne peuvent pas être remplies ou en cas de timeout. L'erreur générée est « prise » dans l'instruction catch. Vous pouvez envelopper des blocs de code plus petits dans une instruction try/catch pour gérer les erreurs à un niveau plus granulaire.

Les développeurs de macros peuvent également générer des erreurs à l'aide de l'instruction

```
'throw new Error('Helpful error message');
```

## Informations supplémentaires

[Objets de l'API Macro](#)[Exemples de macros](#)

## Objets de l'API Macro

---

Vous pouvez créer des macros à l'aide de l'API Macro. Par défaut, quatre objets principaux sont disponibles pour les macros :

- **Session** : point d'entrée principal pour accéder à l'hôte. Vous utilisez l'objet Session pour vous connecter, pour vous déconnecter et pour donner accès à l'objet PresentationSpace.
- **PresentationSpace** : représente l'écran et offre de nombreuses fonctionnalités courantes telles que l'obtention et la définition de l'emplacement du curseur, l'envoi de données à l'hôte et la lecture à partir de l'écran. Vous pouvez l'obtenir en appelant `session.getPresentationSpace()`.
- **Wait** : permet simplement d'attendre que plusieurs états d'hôte se produisent avant de continuer à envoyer davantage de données ou à lire à partir de l'écran. Par exemple, vous pouvez attendre que le curseur se trouve à une certaine position ou que du texte soit présent à une position à l'écran, ou simplement attendre un certain temps. Tous les appels de la fonction « Wait » nécessitent le mot-clé « yield », qui est expliqué ci-dessous.
- **User Interface** : disponible automatiquement dans votre macro en tant que variable « ui ». Cet objet offre des fonctionnalités d'interface utilisateur de base. Vous pouvez l'utiliser pour afficher des données à l'utilisateur ou lui demander de fournir des informations. Tous les appels de la fonction « UI » nécessitent le mot-clé « yield ».

### Ensemble des objets disponibles

Consultez la liste des objets disponibles dans le volet de navigation de droite, « Sur cette page ». (Vous devrez peut-être développer votre navigateur.)

---

## Attribut

Utilisez l'objet Attribute, ainsi que l'objet AttributeSet, pour décoder les informations de mise en forme présentes sur la cellule de données.

Attribut	Description
PROTECTED	Cellule de données protégée
MODIFIED	Cellule de données modifiée

<b>Attribut</b>	<b>Description</b>
NUMERIC_ONLY	Début d'une cellule de données numérique uniquement
ALPHA_NUMERIC	Cellule de données alphanumérique
HIGH_INTENSITY	La cellule de données contient du texte à haute intensité
HIDDEN	La cellule de données contient du texte masqué
PEN_DETECTABLE	La cellule de données peut être détectée par le stylet
ALPHA_ONLY	Cellule de données alphabétique uniquement
NUMERIC_SHIFT	Début d'un champ de décalage numérique
NUMERIC_SPECIAL	La cellule de données marque le début d'un champ numérique spécial
KATAKANA_SHIFT	Section de texte Katakana
MAGNETIC_STRIPE	La cellule de données marque le début d'un champ de bande magnétique
SIGNED_NUMERIC_ONLY	La cellule de données est un champ numérique signé
TRANSMIT_ONLY	La cellule de données est un champ de transmission uniquement
FIELD_END_MARKER	La cellule de données marque la fin d'un champ modifié
FIELD_START_MARKER	La cellule de données marque le début d'un champ modifié
SPECIAL_EMPHASIS_PROTECTED	Champ protégé à mise en valeur spéciale
TAB_STOP	La cellule de données contient un taquet de tabulation
REVERSE	La cellule de données s'affiche en mode vidéo inversée
BLINKING	La cellule de données contient du texte clignotant
RIGHT_JUSTIFIED	La cellule de données marque le début d'un champ justifié à droite
LEFT_JUSTIFIED	La cellule de données marque le début d'un champ justifié à gauche
LOW_INTENSITY	La cellule de données contient du texte de faible intensité



Attribut	Description
UNDERLINE	La cellule de données contient du texte souligné
DOUBLE_BYTE	La cellule de données contient du texte à double octet
COLUMN_SEPARATOR	La cellule de données contient un séparateur de colonnes
BOLD	La cellule de données contient du texte en gras
DOUBLE_WIDTH	La cellule de données marque un champ à double largeur
DOUBLE_HEIGHT_TOP	Cellule de données de niveau supérieur à double hauteur
DOUBLE_HEIGHT_BOTTOM	Cellule de données de niveau inférieur à double hauteur
CONTROL_PAGE_DATA	La cellule de données contient des données de page de contrôle
RIGHT_COLUMN_SEPARATOR	La cellule de données contient un séparateur de colonnes à droite
LEFT_COLUMN_SEPARATOR	La cellule de données contient un séparateur de colonnes à gauche
UPPERSCORE	La cellule de données contient un tiret haut
STRIKE_THROUGH	La cellule de données contient du texte barré

## AttributeSet

L'objet AttributeSet permet à l'utilisateur de décoder les attributs présents dans la cellule de données. Il renvoie les valeurs définies dans l'objet [Attribute](#) et, lorsque ces objets sont utilisés ensemble, vous pouvez obtenir des informations de mise en forme à partir de la cellule de données.

Méthode	Description
<code>contains(attribute)</code>	<p>Détermine si le jeu contient l'objet <a href="#">Attribute</a> spécifié.</p> <p><b>Paramètres</b></p> <p><code>{Number}</code> Attribut à vérifier.</p> <p><b>Valeur renvoyée</b></p> <p><code>{Boolean}</code> True si l'attribut est dans le jeu.</p>

<code>isEmpty()</code>	Détermine si le jeu d'attributs est vide. <b>Valeur renvoyée</b> {Boolean} True si le jeu est vide.
<code>size()</code>	Indique le nombre d'attributs d'un jeu. <b>Valeur renvoyée</b> {Number} Nombre d'attributs.
<code>toArray()</code>	Convertit le jeu d'attributs interne en tableau. <b>Valeur renvoyée</b> {Number[]} Tableau de valeurs des attributs dans le jeu.
<code>toString()</code>	Convertit le jeu d'attributs interne en chaîne. <b>Valeur renvoyée</b> {String} Noms délimités par un espace des attributs dans le jeu.
<code>forEach(callback, thisArg)</code>	Fonction d'itération de chaque élément du jeu d'attributs. <b>Paramètres</b> {forEachCallback} Rappel pour effectuer l'opération spécifique. Appelé avec le nom de chaque attribut du jeu. {Object} thisArg Pointeur facultatif vers un objet de contexte.
<code>forEachCallback(string, object)</code>	Fonction de rappel fournie par l'utilisateur dans laquelle vous fournissez le comportement à utiliser comme paramètre de rappel pour la méthode forEach. <b>Paramètres</b> {String} Nom de chaîne d'un attribut du jeu d'attributs. {Object} thisArg Pointeur facultatif vers un objet de contexte.

## AutoSignOn

Méthode	Description
---------	-------------

<code>getPassTicket()</code>	<p>Obtient un ticket d'accès à utiliser pour l'ouverture d'une session sur une application de mainframe. Plusieurs tickets d'accès peuvent être demandés à l'aide d'identifiants d'application différents.</p> <p><b>Paramètres</b></p> <p>{String} L'identifiant d'application indique à l'hôte l'application concernée par l'ouverture de session.</p> <p><b>Valeur renvoyée</b></p> <p>{Promise} Méthode exécutée avec la clé de ticket d'accès ou rejetée en cas d'échec de l'opération. Le ticket d'accès obtenu auprès de DCAS ne fonctionne qu'avec la session d'hôte en cours et est valable pendant dix minutes.</p>
<code>sendUserName()</code>	<p>Applique le nom d'utilisateur contenu dans le ticket d'accès au champ situé à l'emplacement actuel du curseur sur l'écran d'hôte actuel. Le nom d'utilisateur doit être envoyé avant le mot de passe. Si le mot de passe est envoyé en premier lieu, le ticket d'accès est invalidé et vous devez en obtenir un autre.</p> <p><b>Paramètres</b></p> <p>{String} passTicketKey obtenu auprès de getPassTicketKey</p> <p><b>Valeur renvoyée</b></p> <p>{Promise} Méthode exécutée si le nom d'utilisateur a bien été envoyé. Rejetée en cas d'échec de l'opération.</p>
<code>sendPassword()</code>	<p>Applique le mot de passe contenu dans le ticket d'accès au champ situé à l'emplacement actuel du curseur sur l'écran d'hôte actuel. Le nom d'utilisateur doit être envoyé avant le mot de passe. Si le mot de passe est envoyé en premier lieu, le ticket d'accès est invalidé et vous devez en obtenir un autre.</p> <p><b>Paramètres</b></p> <p>{String} passTicketKey obtenu auprès de getPassTicketKey</p> <p><b>Valeur renvoyée</b></p> <p>{Promise} Méthode exécutée si le mot de passe a bien été envoyé. Rejetée en cas d'échec de l'opération.</p>

## Color

Constantes de couleurs à utiliser pour les couleurs de premier plan et d'arrière-plan de l'objet DataCell.

Couleur	Description	Valeur numérique
BLANK_UNSPECIFIED	Aucune couleur spécifiée	0
BLUE	Bleu	1

Couleur	Description	Valeur numérique
GREEN	Vert	2
CYAN	Cyan	3
RED	Rouge	4
MAGENTA	Magenta	5
YELLOW	Jaune	6
WHITE_NORMAL_INTENSITY	Blanc d'intensité normale	7
GRAY	Gris	8
LIGHT_BLUE	Bleu clair	9
LIGHT_GREEN	Vert clair	10
LIGHT_CYAN	Cyan clair	11
LIGHT_RED	Rouge clair	12
LIGHT_MAGENTA	Magenta clair	13
BLACK	Noir	14
WHITE_HIGH_INTENSITY	Blanc de haute intensité	15
BROWN	Brun	16
PINK	Rose	17
TURQUOISE	Turquoise	18

---

## ControlKey

L'objet ControlKey définit des constantes pour l'envoi de touches de contrôle du curseur et de commandes d'hôte à l'aide de la méthode sendKeys. Les constantes sont disponibles pour les types d'hôtes suivants :

[IBM 3270](#)

[IBM 5250](#)

[VT](#)

## UTS

**IBM 3270**

<b>Mot-clé</b>	<b>Description</b>
ALTVIEW	Autre vue
ATTN	Attention
BACKSPACE	Retour arrière
BACKTAB	Tabulation arrière
CLEAR	Effacer ou effacer l'affichage
CURSOR_SELECT	Curseur de sélection
DELETE_CHAR	Supprimer, supprimer le caractère
DELETE_WORD	Supprimer le mot
DEST_BACK	Retour arrière destructif
DEV_CANCEL	Annulation du périphérique
DOWN	Curseur vers le bas
DSPSOSI	Affichage SO/SI
DUP	Dupliquer le champ
END_FILE	Fin de champ
ENTER	Entrée
ERASE_EOF	Effacer la fin de champ
ERASE_FIELD	Effacer le champ
ERASE_INPUT	Effacer la saisie
FIELD_MARK	Marque de champ
HOME	Origine du curseur
IDENT	Retrait

<b>Mot-clé</b>	<b>Description</b>
INSERT	Insertion
LEFT_ARROW	Curseur vers la gauche
LEFT2	Deux positions vers la gauche
NEW_LINE	Nouvelle ligne
PA1 - PA3	PA1 - PA3
PF1 - PF24	PF1 - PF24
PAGE_DOWN	Page suivante
PAGE_UP	Page précédente
RESET	Réinitialiser, réinitialiser le terminal
RIGHT2	Deux positions vers la droite
RIGHT_ARROW	Curseur vers la droite, droite
SYSTEM_REQUEST	Requête système
TAB	Touche de tabulation
UP	Curseur vers le haut

## IBM 5250

<b>Mot-clé</b>	<b>Description</b>
ALTVIEW	Autre vue
ATTN	Attention
AU1 - AU16	AU1 - AU16
BACKSPACE	Retour arrière
BACKTAB	Tabulation arrière
BEGIN_FIELD	Début de champ
CLEAR	Effacer ou effacer l'affichage

<b>Mot-clé</b>	<b>Description</b>
DELETE_CHAR	Supprimer, supprimer le caractère
DEST_BACK	Retour arrière destructif
DOWN	Curseur vers le bas
DSPSOSI	Affichage SO/SI
DUP	Dupliquer le champ
END_FILE	Fin de champ
ENTER	Entrée
ERASE_EOF	Effacer la fin de champ
ERASE_FIELD	Effacer le champ
ERASE_INPUT	Effacer la saisie
FIELD_EXT	Sortie de champ
FIELD_MINUS	Champ moins
FIELD_PLUS	Champ plus
FIELD_MARK	Marque de champ
HELP	Demande d'aide
HEXMODE	Mode hexadécimal
HOME	Origine du curseur
INSERT	Insertion
LEFT_ARROW	Curseur vers la gauche
NEW_LINE	Nouvelle ligne
PA1 - PA3	PA1 - PA3
PF1 - PF24	PF1 - PF24
PAGE_DOWN	Page suivante
PAGE_UP	Page précédente

<b>Mot-clé</b>	<b>Description</b>
[print]	Imprimer
RESET	Réinitialiser, réinitialiser le terminal
RIGHT_ARROW	Curseur vers la droite, droite
SYSTEM_REQUEST	Requête système
TAB	Touche de tabulation
UP	Curseur vers le haut

## VT

<b>Mot-clé</b>	<b>Description</b>
BACKSPACE	Retour arrière
BREAK	Attn
CLEAR	Effacer ou effacer l'affichage
CURSOR_SELECT	Curseur de sélection
DELETE_CHAR	Supprimer, supprimer le caractère
DOWN	Curseur vers le bas
EK_FIND	Pavé numérique d'édition, rechercher
EK_INSERT	Pavé numérique d'édition, insérer
EK_NEXT	Pavé numérique d'édition, suivant
EK_PREV	Pavé numérique d'édition, précédent
EK_REMOVE	Pavé numérique d'édition, supprimer
EK_SELECT	Pavé numérique d'édition, sélectionner
END_FILE	Fin de champ
ENTER	Entrée
F1 - F24	F1 - F24



<b>Mot-clé</b>	<b>Description</b>
HOLD	Mettre en attente
HOME	Origine
INSERT	Insertion
KEYPAD_COMMA	Virgule sur pavé numérique
KEYPAD_DOT	Décimale sur pavé numérique
KEYPAD_ENTER	Entrée sur pavé numérique
KEYPAD_MINUS	Moins sur pavé numérique
KEYPAD0 - KEYPAD9	Pavé numérique 0 - Pavé numérique 9
LEFT_ARROW	Curseur vers la gauche
PF1 - PF20	PF1 - PF20
PAGE_DOWN	Page suivante
PAGE_UP	Page précédente
RESET	Réinitialiser, réinitialiser le terminal
RETURN	Retour, retour chariot
RIGHT_ARROW	Curseur vers la droite, droite
TAB	Touche de tabulation
UDK16 - UDK20	Touche définie par l'utilisateur 6 - Touche définie par l'utilisateur 20
UP	Curseur vers le haut

## UTS

<b>Mot-clé</b>	<b>Description</b>
BACKSPACE	Retour arrière
BACKTAB	Tabulation arrière
CHAR_ERASE	Efface le caractère au niveau du curseur et fait avancer le curseur

<b>Mot-clé</b>	<b>Description</b>
CLEAR_DISPLAY	Effacer l'affichage
CLEAR_EOD	Effacer jusqu'à la fin de l'affichage
CLEAR_EOF	Effacer jusqu'à la fin du champ
CLEAR_EOL	Effacer jusqu'à la fin de la ligne
CLEAR_FCC	Effacer le caractère de contrôle de champ
CLEAR_HOME	Effacer l'affichage et placer le curseur à l'origine
CONTROL_PAGE	Active/désactive la page de contrôle
DELETE_LINE	Supprime la ligne contenant le curseur et décale les lignes restantes d'une ligne vers le haut
DELIN_LINE	Supprime le caractère placé sous le curseur et décale les caractères restants d'une ligne vers la gauche
DELIN_PAGE	Supprime le caractère placé sous le curseur et décale les caractères restants d'une page vers la gauche
DOWN	Déplace le curseur d'une ligne vers le bas et continue automatiquement dans le bas
DUP_LINE	Crée une copie de la ligne actuelle et remplace la ligne suivante sur la ligne par le doublon
END_FIELD	Déplace le curseur à la fin du champ actuel
END_PAGE	Déplace le curseur à la fin de la page actuelle
EURO	Insère le caractère Euro
F1 - F22	Touches de fonction F1 à F22
HOME	Déplace le curseur au début de la page actuelle (ligne 1, colonne 1)
INSERT	Active/désactive le mode d'insertion/de remplacement.
INSERT_IN_LINE	Insère un espace à la position du curseur et décale les caractères restants d'une ligne vers la droite. Le caractère situé dans la colonne à l'extrême droite de la ligne est rejeté

<b>Mot-clé</b>	<b>Description</b>
INSERT_IN_PAGE	Insère un espace à la position du curseur et décale les caractères restants d'une page vers la droite. Le caractère situé dans la colonne à l'extrême droite de chaque ligne est rejeté
INSERT_LINE	Insère une nouvelle ligne au niveau de la ligne du curseur et décale les lignes restantes vers le bas. La dernière ligne de la page est rejetée
LEFT_ARROW	Déplace le curseur d'une position vers la gauche et continue automatiquement si nécessaire
LOCATE_FCC	Recherche le caractère de contrôle de champ suivant à l'écran
MSG_WAIT	Récupère les messages en file d'attente sur le terminal
RETURN	Retour chariot
RIGHT_ARROW	Déplace le curseur d'une position vers la droite et continue automatiquement si nécessaire
SOE	Insère le caractère de début de saisie
START_OF_FIELD	Déplace le curseur au début du champ
START_OF_LINE	Déplace le curseur vers la colonne 1 de la ligne actuelle
TAB	Déplace le curseur vers la position de tabulation suivante de l'écran
TOGGLE_COL_SEP	Active/désactive l'attribut de séparateur de colonnes
TOGGLE_STRIKE_THRU	Active/désactive l'attribut barré sur la cellule de données actuelle
TOGGLE_UNDERLINE	Active/désactive l'attribut souligné sur la cellule de données actuelle
TRANSMIT	Transmet les données de champ modifiées à l'hôte
UNLOCK	Envoie la touche UNLOCK à l'hôte

Mot-clé	Description
UP	Déplace le curseur d'une ligne vers le haut et continue automatiquement si nécessaire

## DataCell

L'objet DataCell fournit des informations sur une position particulière sur un écran de terminal.

Méthode	Description
<code>getPosition()</code>	Renvoie la position de cette cellule de données à l'écran. <b>Valeur renvoyée</b> {Position} Position de la cellule de données à l'écran.
<code>getChar()</code>	Obtient le caractère associé à la cellule. <b>Valeur renvoyée</b> {String} Caractère associé à la cellule.
<code>getAttributes()</code>	Renvoie le jeu d'attributs spécifié pour cette instance de cellule de données. Voir <a href="#">AttributeSet</a> . <b>Valeur renvoyée</b> {AttributeSet} Jeu d'attributs pour cette instance de cellule de données.
<code>getForegroundColor()</code>	Renvoie la couleur de premier plan, telle qu'elle est définie dans l'objet Color, pour cette cellule de données. <b>Valeur renvoyée</b> {Number} Couleur de premier plan pour cette cellule de données. La couleur est définie dans l'objet <a href="#">Color</a> .
<code>getBackgroundColor()</code>	Renvoie la couleur d'arrière-plan, telle qu'elle est définie dans l'objet Color, pour cette cellule de données. <b>Valeur renvoyée</b> {Number} Couleur d'arrière-plan pour cette cellule de données. La couleur est définie dans l'objet <a href="#">Color</a> .
<code>toString</code>	Convertit la cellule de données interne en chaîne. <b>Valeur renvoyée</b> {String} Représentation sous forme de chaîne d'une cellule de données.

<code>isFieldDelimiter</code> <code>()</code>	<p>Teste si cette cellule représente un séparateur de champs.</p> <p><b>Valeur renvoyée</b></p> <p><code>{Boolean}</code> True si cette cellule est un séparateur de champs, False dans le cas contraire.</p>
--	---

## Dimension

Représente la taille de l'écran ou de la zone d'écran.

Méthode	Description
<code>Dimension(rows, cols)</code>	<p>Crée une instance Dimension.</p> <p><b>Paramètres</b></p> <p><code>{Number} rows</code> Dimension de l'écran en lignes.</p> <p><code>{Number} cols</code> Dimension de l'écran en colonnes.</p>

## Field

Utilisez l'objet Field, avec l'objet [FieldList](#), pour obtenir les informations présentes dans un champ à l'écran.

Méthode	Description
<code>getAttributes()</code>	<p>Renvoie le jeu d'attributs spécifié pour cette instance de champ. Voir <a href="#">AttributeSet</a>.</p> <p><b>Valeur renvoyée</b></p> <p><code>{AttributeSet}</code> Jeu d'attributs pour ce champ.</p>
<code>getForegroundColor()</code>	<p>Renvoie la couleur de premier plan du champ.</p> <p><b>Valeur renvoyée</b></p> <p><code>{Number}</code> Couleur de premier plan pour ce champ. Cette valeur est définie dans l'objet <a href="#">Color</a>.</p>
<code>getBackgroundColor()</code>	<p>Renvoie la couleur d'arrière-plan du champ.</p> <p><b>Valeur renvoyée</b></p> <p><code>{Number}</code> Couleur d'arrière-plan pour ce champ. Cette valeur est définie dans l'objet <a href="#">Color</a>.</p>

getStart()	<p>Renvoie la position de début du champ. La position de début est la position du premier caractère du champ. Certains types d'hôtes utilisent une position de caractère pour stocker les attributs de niveau champ. Dans ce cas, la position de l'attribut n'est pas considérée comme position de début.</p> <p><b>Valeur renvoyée</b>  {Position} Position de début du champ.</p> <p><b>Erreur générée</b>  {RangeError} Pour les champs de longueur zéro.</p>
getEnd()	<p>Renvoie la position de fin du champ. La position de fin est la position dans l'espace de présentation qui contient le dernier caractère du champ.</p> <p><b>Valeur renvoyée</b>  {Position} Position de fin du champ.</p> <p><b>Erreur générée</b>  {RangeError} Pour les champs de longueur zéro.</p>
getLength() )	<p>Renvoie la longueur du champ. Pour les types d'hôtes qui utilisent une position de caractère pour stocker les attributs de champ, la longueur du champ n'inclut pas la position de l'attribut de champ.</p> <p><b>Valeur renvoyée</b>  {Number} Longueur du champ.</p>
getDataCells()	<p>Obtient les cellules de données qui composent ce champ. Voir <a href="#">DataCell</a>.</p> <p><b>Valeur renvoyée</b>  {DataCell[]} Cellules de données qui composent ce champ.</p>
getText()	<p>Obtient le texte dans le champ.</p> <p><b>Valeur renvoyée</b>  {String} Texte du champ.</p>
setText()	<p>Définit le texte du champ. Pour certains types d'hôtes, comme VT, le texte est transmis immédiatement à l'hôte. Toutefois, pour d'autres types d'hôtes, le texte n'est transmis à l'hôte que si une touche Aid est appelée. Si le texte est plus court que le champ, il est placé dans le champ d'hôte, et le reste du champ est effacé. Si le texte est plus long que le champ d'hôte, le plus de texte possible est placé dans le champ.</p> <p><b>Paramètre</b>  {String} Texte à définir dans le champ.</p> <p><b>Erreur générée</b>  {Error} Si le champ est protégé.</p>

<code>clearField()</code>	Efface le champ actuel d'une manière spécifique à l'émulation. <b>Erreur générée</b> {Error} Si le champ est protégé, ou effacer dans le cas contraire.
<code>getPresentationSpace()</code>	Obtient l'objet PresentationSpace qui a créé ce champ. <b>Valeur renvoyée</b> {PresentationSpace} Parent de cette instance de champ.
<code>toString()</code>	Crée une description conviviale du champ. <b>Valeur renvoyée</b> {String} Version du champ lisible par l'utilisateur.

## FieldList

Utilisez l'objet FieldList, ainsi que l'objet Field, pour obtenir des informations sur la liste de champs.

Méthode	Description
<code>getPresentationSpace()</code>	Obtient l'objet <a href="#">PresentationSpace</a> qui a créé ce champ. <b>Valeur renvoyée</b> {PresentationSpace} Parent de cette instance de champ.

<code>findField( position, text, direction)</code>	<p>Renvoie le champ contenant le texte spécifié. La recherche commence à partir de la position spécifiée et se poursuit vers l'avant ou vers l'arrière. Si la chaîne s'étend sur plusieurs champs, le champ contenant la position de début est renvoyé. Lors d'une recherche vers l'avant, la recherche ne se poursuit pas automatiquement en haut de l'écran. Lors d'une recherche vers l'arrière, la recherche ne se poursuit pas en bas de l'écran.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li><code>{Position}</code> Position de début de la recherche. Voir l'objet <code>Position</code>.</li> <li><code>{String}</code> Texte à rechercher (facultatif). S'il n'est pas fourni, renvoie le champ suivant à droite ou en dessous de la position spécifiée.</li> <li><code>{Number}</code> Direction de la recherche (facultatif). Utilisez des constantes <code>PresentationSpace</code>. <code>SearchDirection</code> pour ce paramètre. Par exemple, <code>PresentationSpace.SearchDirection.FORWARD</code> ou <code>PresentationSpace.SearchDirection.BACKWARD</code>. Si elle n'est pas fournie, la recherche est effectuée vers l'avant.</li> </ul> <p><b>Valeur renvoyée</b></p> <ul style="list-style-type: none"> <li><code>{Field}</code> Contient la chaîne ou la valeur null si aucun champ répondant aux critères donnés n'est trouvé.</li> </ul> <p><b>Erreur générée</b></p> <ul style="list-style-type: none"> <li><code>{RangeError}</code> Si la position est hors plage.</li> </ul>
<code>get(index)</code>	<p>Obtient le champ à l'index donné.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"> <li><code>{Number}</code> Index dans la liste de champs.</li> </ul> <p><b>Valeur renvoyée</b></p> <ul style="list-style-type: none"> <li><code>{Field}</code> Située dans l'index spécifié.</li> </ul> <p><b>Erreur générée</b></p> <ul style="list-style-type: none"> <li><code>{RangeError}</code> Si l'index est hors plage.</li> </ul>
<code>isEmpty()</code>	<p>Détermine si la liste de champs est vide.</p> <p><b>Valeur renvoyée</b></p> <ul style="list-style-type: none"> <li><code>{Boolean}</code> True si la liste est vide.</li> </ul>
<code>size()</code>	<p>Indique le nombre de champs de la liste.</p> <p><b>Valeur renvoyée</b></p> <ul style="list-style-type: none"> <li><code>{Number}</code> Nombre de champs.</li> </ul>



`toString()` Crée une description conviviale de la liste de champs.  
**Valeur renvoyée**  
`{String}` Version de la liste de champs lisible par l'utilisateur.

## FileTransfer

Utilisez l'objet FileTransfer pour répertorier et transférer des fichiers entre le système hôte et le client.

L'API de transfert de fichiers Host Access for the Cloud exploite les conventions de chemin d'accès aux fichiers utilisées par différentes implémentations de fichiers d'hôte. Suivez les formats d'URL ou de chemin d'accès au système de fichiers Linux pour mettre en forme les chemins d'accès aux fichiers utilisés par l'API. Par exemple, `/racine/répertoire/fichier`.

Il est important de respecter toutes les règles spécifiques des systèmes hôtes, telles que les caractères autorisés ou les longueurs de noms.

### 💡 remarque

Les navigateurs imposent des restrictions de sécurité importantes en ce qui concerne la capacité de JavaScript à interagir avec les systèmes de fichiers client.

Méthode	Description
<code>getHostFileListing(remotePath)()</code>	<p>Demande une liste des fichiers d'hôte. Si <code>RemotePath</code> est omis, la liste de fichiers du répertoire de travail distant actuel s'affiche.</p> <p><b>Paramètres</b></p> <p><code>{String}</code> (Facultatif) Si ce paramètre est spécifié, il permet d'obtenir la liste de fichiers pour le chemin d'accès distant spécifié. Dans le cas contraire, vous obtenez la liste de fichiers du répertoire de travail distant actuel.</p> <p><b>Valeur renvoyée</b></p> <p><code>{Promise}</code> Correspond à un tableau d'objets HostFile contenus dans <code>remoteName</code>. Méthode rejetée si le chemin d'accès distant ne peut pas être lu.</p>

`sendFile(localFile, remoteName)`

Envoie le fichier spécifié à l'hôte.

**Paramètres**

`{File}` Objet Fichier JavaScript pointant vers le fichier local à envoyer.  
`{String}` (Facultatif) Nom complet du fichier distant tel qu'il est autorisé par le système distant (Unix, Windows, MVS, VAX).

**Valeur renvoyée**

`{Promise}` Méthode exécutée avec un objet `HostFile` représentant le fichier envoyé en cas de réussite. Méthode rejetée si une erreur se produit lors de l'envoi du fichier.

`getDownloadURL(remoteName)`

Construit un lien pour télécharger un fichier à partir d'un système hôte.

**Paramètres**

`{String}` Nom complet du fichier distant tel qu'il est autorisé par le système distant (Unix, Windows, MVS, VAX).

**Valeur renvoyée**

`{URL}` URL qui peut être utilisée pour récupérer le fichier à partir du serveur de session Host Access for the Cloud.

`setTransferOptions(options)`

Définit les options de transfert pour la session FileTransfer actuelle. Les options de transfert sont appliquées à tous les transferts ultérieurs jusqu'à ce que la session soit fermée ou remplacée par un autre appel de `setTransferOptions`.

**Paramètres**

`{JSON}` Voir `FileTransferOptions` pour les noms et valeurs autorisés.

**Valeur renvoyée**

`{Promise}` Méthode exécutée à la fin de l'appel. Rejetée en cas d'erreur lors de la définition des options.

`cancel()`

Annule le transfert en cours.

**Paramètres**`{String}` Nom complet du fichier distant tel qu'il est autorisé par le système distant (Unix, Windows, MVS, VAX).**Valeur renvoyée**`{Promise}` Méthode exécutée à la fin de l'appel. Rejetée si une erreur se produit lors de l'annulation du transfert.

## FileTransferFactory

Un objet FileTransfer Factory est disponible pour toutes les macros. Si des transferts de fichiers sont configurés pour la session, vous pouvez l'utiliser pour obtenir une référence à un objet FileTransfer.

Méthode	Description
<code>getIND\$File()</code>	<p>Renvoie un objet FileTransfer pour interagir avec le type Ind\$File configuré pour la session.</p> <p><b>Valeur renvoyée</b> <code>{FileTransfer}</code></p> <p><b>Erreur générée</b> <code>{Error}</code> Si la session n'a pas été configurée pour autoriser les transferts IND\$File.</p>

## FileTransferOptions

Spécification de l'objet des options de transfert de fichiers. Exemple :

```
`fileTransfer.setTransferOptions({ transferMethod : 'ascii' });``
```

Méthode	Description
---------	-------------

```
transferMethod    {String} Valeurs autorisées :
                  'ascii'
                  'binary'
```

## HostFile

Un objet HostFile représente un fichier sur le système de fichiers hôte.

Méthode	Description
<code>getName()</code>	Obtient le nom du fichier. <b>Valeur renvoyée</b> <code>{String}</code> Nom du fichier.
<code>getParent()</code>	Obtient le parent de ce fichier d'hôte. <b>Valeur renvoyée</b> <code>{String}</code> Parent de ce fichier d'hôte. La signification varie selon le type d'hôte. Par exemple, sur TSO, il s'agit du nom du catalogue dans lequel réside le fichier.
<code>getSize()</code>	Taille du fichier, en octets. <b>Valeur renvoyée</b> <code>{Number}</code> Taille du fichier en octets.
<code>getType()</code>	Type de fichier représenté. <b>Valeur renvoyée</b>

## HostFileType

L'objet HostFileType définit des constantes pour déterminer le type d'un objet HostFile.

Valeur	Description
FILE	Représente un fichier sur le système hôte.
DIR	Représente un répertoire sur le système hôte.

Valeur	Description
UNKNOWN	Représente un fichier d'hôte d'origine inconnue.

## OIA

Interface OIA (Operator Information Area). L'objet OIA renvoie les valeurs définies dans l'objet OIAStatus.

Méthode	Description
<code>getStatus ()</code>	<p>Renvoie le jeu de drapeaux d'état activés. Voir <a href="#">StatusSet</a>.</p> <p><b>Valeur renvoyée</b>  <code>{StatusSet}</code> Contient les informations d'état.</p>
<code>getCommErrorCode ()</code>	<p>Renvoie le code d'erreur de communication actuel.</p> <p><b>Valeur renvoyée</b>  <code>{Number}</code> Code d'erreur de communication actuel. S'il n'en existe pas, la valeur est 0.</p>
<code>getProgErrorCode ()</code>	<p>Renvoie le code d'erreur de programme actuel.</p> <p><b>Valeur renvoyée</b>  <code>{Number}</code> Code d'erreur de programme actuel. S'il n'en existe pas, la valeur est 0.</p>

## OIAStatus

OIAStatus	Description
CONTROLLER_READY	Contrôleur prêt
A_ONLINE	En ligne avec une connexion non-SNA
MY_JOB	Connecté à une application hôte
OP_SYS	Connecté à un SSCP (SNA)
UNOWNED	Non connecté
TIME	Clavier inhibé

<b>OIAStatus</b>	<b>Description</b>
SYS_LOCK	Verrouillage système après la touche AID
COMM_CHECK	Vérification de la communication
PROG_CHECK	Vérification du programme
ELSEWHERE	Touche non valide à l'emplacement du curseur
FN_MINUS	Fonction non disponible
WHAT_KEY	Touche non valide
MORE_THAN	Trop de caractères entrés dans le champ
SYM_MINUS	Le symbole entré n'est pas disponible
INPUT_ERROR	Erreur de saisie de l'opérateur (5250 uniquement)
DO_NOT_ENTER	Ne pas entrer
INSERT	Curseur en mode d'insertion
GR_CURSOR	Curseur en mode graphique
COMM_ERR_REM	Rappel d'erreur de communication
MSG_WAITING	Indicateur de message en attente
ENCRYPT	La session est codée
NUM_FIELD	Caractère non valide dans le champ numérique uniquement

## Position

<b>Méthode</b>	<b>Description</b>

`Position(row,col)`

Crée une instance Position.

**Paramètres**`{Number}` Coordonnées de la ligne à l'écran.`{Number}` Coordonnées de la colonne à l'écran.

## PresentationSpace

Utilisez l'objet PresentationSpace pour interagir avec l'écran du terminal. La définition et l'obtention de la position du curseur, l'envoi de touches et la lecture de texte sont quelques-unes des interactions disponibles.

Méthode	Description
<code>getCursorPosition()</code>	<p>Renvoie une instance <a href="#">Position</a> représentant la position actuelle du curseur. Pour une session non connectée, la position du curseur est 0,0.</p> <p><b>Valeur renvoyée</b> <code>{Position}</code> Emplacement actuel du curseur.</p>
<code>setCursorPosition(position)</code>	<p>Déplace le curseur d'hôte vers la ligne et la colonne spécifiées. Pour certains hôtes, tels que VT, l'hôte peut limiter les mouvements du curseur.</p> <p><b>Paramètres</b> <code>{Position}</code> <a href="#">Position</a> Nouvelle position du curseur.</p> <p><b>Valeur renvoyée</b> Aucune.</p> <p><b>Erreur générée</b> <code>{RangeError}</code> Si la position n'est pas valide sur l'écran actuel.</p>
<code>isCursorVisible()</code>	<p>Teste que le curseur est actuellement visible dans l'espace de présentation. Le curseur est considéré comme n'étant pas visible si la session n'est pas connectée.</p> <p><b>Valeur renvoyée</b> <code>{Boolean}</code> True si le curseur est visible. False si le curseur n'est pas visible.</p>

`sendKeys (keys  
)`

Transmet une chaîne de texte ou un objet `ControlKey` à l'hôte à la position actuelle du curseur dans l'espace de présentation. Si le curseur ne se trouve pas à la position souhaitée, utilisez d'abord la fonction `setCursorPosition`.

La chaîne de texte peut contenir un nombre indéfini de caractères et d'objets `ControlKey`.

Par exemple : "nom" + `ControlKey.TAB` + "mot\_passe" + `ControlKey.ENTER` transmet un identifiant d'utilisateur, passe au champ suivant, transmet un mot de passe, puis transmet la touche Entrée.

Si vous devez transmettre un crochet, doublez-le ([[ ou ]]).

#### Paramètres

`{String}` Texte et/ou touches de contrôle à transmettre.

`getText (start  
, length)`

Renvoie une chaîne représentant une zone linéaire de l'espace de présentation. Aucun caractère de nouvelle ligne n'est inséré si des limites de ligne sont rencontrées.

#### Paramètres

`{Position}` Position de début de la récupération de texte.

`{Number}` Nombre maximal de caractères à renvoyer. Si le paramètre de longueur dépasse la dernière position de l'espace de présentation, seuls les caractères jusqu'à cette dernière position sont renvoyés.

#### Valeur renvoyée

`{String}` Représente une zone linéaire de l'espace de présentation qui peut être vide si la session n'est pas connectée.

#### Erreur générée

`{RangeError}` Si la position ou la longueur n'est pas valide sur l'écran actuel.

`getSize()`

Obtient les dimensions de l'écran en tant qu'objet `Dimension`.

#### Valeur renvoyée

`{Dimension}` Contient le nombre de lignes et de colonnes. La taille de l'écran est [row:0, col:0] si la session n'est pas connectée.



`getDataCells (start, length)` Renvoie les instances [DataCell](#) dans lesquelles le premier membre correspond à la position spécifiée par le paramètre de début. Le nombre maximal d'instances DataCell dans la liste est spécifié par le paramètre de longueur.

**Paramètres**

`{Position}` Début de la première position de l'écran d'hôte dans laquelle récupérer les instances DataCell. Voir [Position](#).

`{Number}` Nombre maximal d'instances DataCell à récupérer. Si cette valeur n'est pas spécifiée, renvoie les instances DataCell de la position de début jusqu'à la fin de l'écran.

**Valeur renvoyée**

`{DataCell[]}` Instances qui peuvent être vides si la session n'est pas connectée. Si la position n'est pas spécifiée, renvoie toutes les instances DataCell. Si la longueur n'est pas spécifiée, renvoie les instances DataCell de la position de début jusqu'à la fin de l'écran.

**Erreur générée**

`{RangeError}` Si le début ou la longueur est hors plage.

`getFields()` Renvoie la liste des champs de l'espace de présentation. Si le type d'hôte ne prend pas en charge les champs ou si l'écran actuel n'est pas mis en forme, la valeur renvoyée est toujours une liste vide. Voir [FieldList](#).

**Valeur renvoyée**

`{FieldList}` Liste des champs définis par l'hôte dans l'espace de présentation.

## Session

L'objet Session est le point d'entrée principal pour interagir avec l'hôte. Il contient des fonctions de connexion, de déconnexion et d'obtention de l'objet PresentationSpace.

Méthode	Description
<code>connect()</code>	<p>Se connecte à l'hôte configuré. Si nécessaire, utilisez <code>wait.forConnect()</code> pour bloquer l'exécution des macros jusqu'à ce que la session soit connectée.</p> <p><b>Valeur renvoyée</b></p> <p>Aucune</p>

<code>disconnect()</code>	<p>Se déconnecte de l'hôte configuré. Si nécessaire, utilisez <code>wait.forDisconnect()</code> pour bloquer l'exécution des macros jusqu'à ce que la session soit connectée.</p> <p><b>Valeur renvoyée</b> Aucune</p>
<code>isConnected()</code>	<p>Détermine si la connexion à l'hôte est connectée.</p> <p><b>Valeur renvoyée</b> {Boolean} True si la connexion à l'hôte est établie ; False si ce n'est pas le cas.</p>
<code>getPresentationSpace()</code>	<p>Permet d'accéder à l'instance PresentationSpace pour cette session.</p> <p><b>Valeur renvoyée</b> {PresentationSpace} Instance associée à cette session.</p>
<code>getDeviceName()</code>	<p>Renvoie le nom du périphérique pour une session connectée ou une chaîne vide si la session est déconnectée ou si elle n'a pas de nom de périphérique.</p> <p><b>Valeur renvoyée</b> {String} Nom du périphérique connecté.</p>
<code>getType()</code>	<p>Renvoie le type de session d'hôte. Voir <a href="#">SessionType</a>.</p> <p><b>Valeur renvoyée</b> {String} Type de session d'hôte.</p>
<code>setDeviceName()</code>	<p>Permet de modifier le nom du périphérique sur une instance de session.</p> <p><b>Paramètres</b> {String} Nom du périphérique à utiliser lors de la connexion à un hôte.</p> <p><b>Erreur générée</b> {Error} En cas de tentative de définition du nom du périphérique alors que la session est connectée.</p>

`getOIA()` Permet d'accéder à l'instance [OIA](#) pour cette session.  
**Valeur renvoyée**  
`{OIA}` Valeur associée à cette session.

## SessionType

Constantes utilisées pour identifier le type d'hôte auquel la connexion est établie. Voir l'objet [Session](#).

Types d'hôtes disponibles :

IBM\_3270

IBM\_5250

VT

ALC

UTS

T27

## StatusSet

Vous pouvez utiliser l'objet StatusSet pour décoder l'état OIA. L'objet StatusSet renvoie les valeurs définies dans l'objet [OIAStatus](#) et, lorsqu'ils sont utilisés ensemble, vous pouvez obtenir les informations d'état de l'OIA.

Méthode	Description
<code>contains(statusFlag)</code>	Détermine si le jeu contient le drapeau d'état spécifié dans les constantes OIAStatus. <b>Paramètres</b> <code>{Number}</code> État statusFlag à vérifier. <b>Valeur renvoyée</b> <code>{Boolean}</code> True si le drapeau d'état est présent dans le jeu.
<code>isEmpty()</code>	Détermine si le jeu d'états est vide. <b>Valeur renvoyée</b> <code>{Boolean}</code> True si le jeu est vide.
<code>size()</code>	Indique le nombre de drapeaux d'état dans le jeu. <b>Valeur renvoyée</b> <code>{Number}</code> Nombre d'états.

<code>toArray()</code>	<p>Convertit le jeu d'états interne en tableau.</p> <p><b>Valeur renvoyée</b></p> <p><code>{Object []}</code> Tableau de drapeaux d'état dans le jeu.</p>
<code>toString()</code>	<p>Convertit le jeu d'états interne en chaîne.</p> <p><b>Valeur renvoyée</b></p> <p><code>{String}</code> Noms délimités par un espace des drapeaux d'état du jeu.</p>
<code>forEach(callback, thisArg)</code>	<p>Fonction à itérer sur chaque élément du jeu d'états.</p> <p><b>Paramètres</b></p> <p><code>{forEachCallback}</code> Rappel pour effectuer l'opération spécifique. Méthode appelée avec le nom de chaque état du jeu.</p>
<code>forEachCallback(string, thisArg)</code>	<p>Fonction de rappel fournie par l'utilisateur dans laquelle vous fournissez le comportement à utiliser comme paramètre de rappel pour la méthode <code>forEach</code>.</p> <p><b>Paramètres</b></p> <p><code>{String} String</code> Nom d'un état dans le jeu d'états.</p> <p><code>{Object} thisArg</code> Pointeur facultatif vers un objet de contexte.</p>

## User Interface

L'objet User Interface offre des fonctions pour interagir avec l'utilisateur, ainsi que pour demander et afficher des informations de base. L'objet UI est automatiquement disponible dans votre macro en tant que variable « ui ».

### remarque

**Important** : toutes les fonctions UI doivent être précédées du mot-clé « yield ». Cela permet à la macro de bloquer l'exécution jusqu'à ce que les conditions de la fonction UI soient respectées.

`[parameter]` indique un paramètre facultatif.

Méthode	Description
---------	-------------

<pre>prompt(message, [defaultAnswer], [mask])</pre>	<p>Invite l'utilisateur à fournir des informations dans l'interface utilisateur.</p> <p><b>Paramètres</b></p> <p>{String} message : titre à afficher à l'utilisateur. Valeur par défaut : chaîne vide.</p> <p>{String} defaultAnswer : réponse à utiliser si l'utilisateur la laisse vide. Valeur par défaut : chaîne vide.</p> <p>{Boolean} mask : indique s'il faut masquer l'invite (comme avec un mot de passe).</p> <p><b>Valeur renvoyée</b></p> <p>{Promise} Méthode exécutée lorsque l'utilisateur ferme la boîte de dialogue. Renvoie l'entrée de l'utilisateur sur « OK » ou null sur « Annuler ».</p>
<pre>message([message])</pre>	<p>Affiche un message dans l'interface utilisateur.</p> <p><b>Paramètres</b></p> <p>{String} Message à afficher à l'utilisateur. Valeur par défaut : chaîne vide.</p> <p><b>Valeur renvoyée</b></p> <p>{Promise} Méthode exécutée lorsque l'utilisateur ferme la fenêtre du message.</p>

## Wait

Utilisez l'objet Wait pour attendre un état de session ou d'écran particulier. Par exemple, vous pouvez attendre que le curseur se trouve à un emplacement particulier ou que du texte soit présent à un emplacement donné avant de poursuivre l'exécution d'une macro.

Les fonctions Wait sont souvent utilisées avec des fonctions asynchrones telles que connect() et sendKeys().

### remarque

Toutes les fonctions utilisent le timeout en tant que paramètre facultatif et ont une valeur de timeout par défaut de 10 secondes (10 000 ms).

**Important** : toutes les fonctions Wait doivent être précédées du mot-clé « yield ». Cela permet à la macro de bloquer l'exécution jusqu'à ce que les conditions de la fonction Wait soient respectées.

[parameter] indique un paramètre facultatif.

Méthode	Description
<code>setDefaultTimeout(timeout)</code>	<p>Définit la valeur de timeout par défaut pour toutes les fonctions.</p> <p><b>Paramètres</b></p> <p><code>{Number}</code> Timeout par défaut à utiliser pour toutes les fonctions Wait en millisecondes.</p> <p><b>Valeur renvoyée</b></p> <p><code>{None}</code></p> <p><b>Erreur générée</b></p> <p><code>{RangeError}</code> Si le timeout spécifié est inférieur à zéro.</p>
<code>forConnect([timeout])</code>	<p>Attend la fin d'une requête de connexion.</p> <p><b>Paramètres</b></p> <p><code>{Number}</code> Timeout en millisecondes.</p> <p><b>Valeur renvoyée</b></p> <p><code>{Promise}</code> Méthode exécutée si la session est déjà connectée ou lors de la connexion. Rejetée en cas de timeout de la période d'attente.</p>
<code>forDisconnect([timeout])</code>	<p>Attend la fin d'une requête de déconnexion.</p> <p><b>Paramètres</b></p> <p><code>{Number}</code> Timeout en millisecondes.</p> <p><b>Valeur renvoyée</b></p> <p><code>{Promise}</code> Méthode exécutée si la session est déjà déconnectée ou lors de la déconnexion. Rejetée en cas de timeout de la période d'attente.</p>
<code>forFixedTime([timeout])</code>	<p>Attend sans condition pendant une période fixe, en millisecondes (ms).</p> <p><b>Paramètres</b></p> <p><code>{Number}</code> Timeout en millisecondes.</p> <p><b>Valeur renvoyée</b></p> <p><code>{Promise}</code> Méthode exécutée à l'issue du délai.</p>
<code>forScreenChange([timeout])</code>	<p>Attend que l'écran d'hôte change. Cette fonction indique quand une mise à jour de l'écran est détectée. Elle ne garantit pas le nombre de mises à jour ultérieures qui peuvent se produire avant la fin de l'écran. Il est conseillé d'attendre que le contenu de l'écran corresponde à certains critères d'arrêt connus.</p> <p><b>Paramètres</b></p> <p><code>{Number}</code> Timeout en millisecondes.</p> <p><b>Valeur renvoyée</b></p> <p><code>{Promise}</code> Méthode résolue si l'écran change. Rejetée en cas de timeout de la période d'attente.</p>

<code>forCursor(position, [timeout])</code>	<p>Attend que le curseur arrive à la position spécifiée.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"><li><code>{Position}</code> Position indiquant la ligne et la colonne.</li></ul> <p><b>Valeur renvoyée</b></p> <ul style="list-style-type: none"><li><code>{Promise}</code> Méthode exécutée si le curseur est déjà localisé ou lorsqu'il est enfin localisé. Rejetée en cas de timeout de la période d'attente.</li></ul>
<code>forText(text, position, [timeout])</code>	<p>Attend du texte situé à une position spécifique de l'écran</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"><li><code>{String}</code> Text à attendre.</li><li><code>{Position}</code> Position indiquant la ligne et la colonne.</li><li><code>{Number}</code> Timeout en millisecondes.</li></ul> <p><b>Valeur renvoyée</b></p> <ul style="list-style-type: none"><li><code>{Promise}</code> Méthode exécutée si le texte se trouve déjà à la position spécifiée ou lorsqu'il est localisé. Rejetée en cas de timeout de la période d'attente.</li></ul> <p><b>Erreur générée</b></p> <ul style="list-style-type: none"><li><code>{RangeError}</code> Si la position n'est pas valide.</li></ul>
<code>forHostPrompt(text, column, [timeout])</code>	<p>Attend une invite de commande située dans une colonne spécifique de l'écran.</p> <p><b>Paramètres</b></p> <ul style="list-style-type: none"><li><code>{String}</code> Invite à attendre.</li><li><code>{Number}</code> Colonne dans laquelle le curseur est attendu.</li><li><code>{Number}</code> Timeout en millisecondes.</li></ul> <p><b>Valeur renvoyée</b></p> <ul style="list-style-type: none"><li><code>{Promise}</code> Méthode exécutée si les conditions sont déjà respectées ou lorsqu'elles sont enfin respectées. Rejetée en cas de timeout de la période d'attente.</li></ul> <p><b>Erreur générée</b></p> <ul style="list-style-type: none"><li><code>{RangeError}</code> Si la colonne est hors plage.</li></ul>

```
forHostSettle(
  [settleTime],
  [timeout])
```

**REMARQUE :** `wait.forHostSettle` ne doit être utilisé que lorsque d'autres fonctions Wait plus ciblées sont insuffisantes.

Surveille les données entrantes de l'écran et résout `settleTime` après la dernière mise à jour **et** le déverrouillage du clavier. Cette fonction est utile lorsque des données arrivent dans plusieurs paquets et que vous souhaitez vous assurer que tout l'écran a été reçu avant de continuer.

#### Paramètres

`{Number}` Durée d'attente après la dernière mise à jour pour s'assurer que d'autres données n'arrivent pas de manière inattendue. La valeur par défaut est de 200 millisecondes.

`{Number}` Timeout en millisecondes.

#### Valeur renvoyée

`{Promise}` Méthode exécutée lorsque la durée de stabilisation est écoulée après la réception de la dernière mise à jour de l'écran et que le clavier est déverrouillé.

## Exemples de macros

---

Pour vous aider à créer des macros qui tirent parti de toutes les fonctionnalités de l'éditeur de macros, les exemples ci-après sont disponibles comme point de départ.

## Interactions de base avec l'hôte

---

Cet exemple illustre les interactions de base avec l'hôte, notamment :

- Envoi de données à l'hôte

- Attente de l'affichage d'écrans

- Utilisation du mot-clé `yield` pour attendre des fonctions asynchrones

- Lecture de texte à l'écran

- Affichage d'informations de base pour l'utilisateur

- Notions de base en matière de gestion des erreurs

Les objets suivants sont disponibles par défaut pour toutes les macros :

1. **session** : point d'entrée principal pour accéder à l'hôte. Il permet de se connecter, de se déconnecter et d'accéder à l'objet `PresentationSpace`.



L'objet `PresentationSpace` obtenu à partir de la session représente l'écran et offre de nombreuses fonctionnalités courantes telles que l'obtention et la définition de l'emplacement du curseur, l'envoi de données à l'hôte et la lecture à partir de l'écran.

2. **wait** : permet simplement d'attendre que plusieurs états d'hôte se produisent avant de continuer à envoyer davantage de données ou à lire à partir de l'écran.
3. **UI** : offre des fonctionnalités d'interface utilisateur de base. Il permet d'afficher des données à l'utilisateur ou de lui demander de fournir des informations.

```

// Créer une fonction de macro
var macro = createMacro(function*(){
  'use strict';

  // Les objets suivants sont disponibles par défaut pour toutes les macros :
  // 1. session : point d'entrée principal pour accéder à l'hôte. Il permet de
  se connecter, de se déconnecter et d'accéder à l'objet PresentationSpace.
  // L'objet PresentationSpace obtenu à partir de la session représente
  l'écran et offre de nombreuses fonctionnalités courantes telles que
  l'obtention et la définition de
  // l'emplacement du curseur, l'envoi de données à l'hôte et la lecture à
  partir de l'écran.
  // 2. wait : permet simplement d'attendre que plusieurs états d'hôte se
  produisent avant de continuer à envoyer davantage de données ou à lire à
  partir de l'écran.
  // 3. ui : offre des fonctionnalités d'interface utilisateur de base. Il
  permet d'afficher des données à l'utilisateur ou de lui demander de fournir
  des informations.

  // Déclarer une variable pour lire et afficher des données d'écran.
  // Comme meilleure pratique, toutes les variables doivent être déclarées au
  début d'une fonction.
  var numberOfAccounts = 0;

  // Commencer par obtenir l'objet PresentationSpace, qui fournit de
  nombreuses opérations d'écran courantes.
  var ps = session.getPresentationSpace();

  try {
    // Permet de définir et d'obtenir l'emplacement du curseur.
    ps.setCursorPosition(new Position(24, 2));

    // Utiliser la fonction sendKeys pour envoyer des caractères à l'hôte.
    ps.sendKeys('cics');

    // La fonction SendKeys est également utilisée pour envoyer des touches
    d'hôte telles que les touches PA et PF.
    // Se reporter à la section sur les touches de contrôle dans la
    documentation pour connaître toutes les options disponibles.
    ps.sendKeys(ControlKey.ENTER);

    // Attendre que le curseur soit à la position correcte.
    // L'objet wait fournit diverses fonctions pour attendre que certains
    états se produisent
    // afin de pouvoir continuer à envoyer d'autres touches ou à lire des
    données de l'écran.
  }
}

```

```

yield wait.forCursor(new Position(24, 2));

// Vous pouvez mélanger des caractères et des touches de contrôle dans
un seul appel sendKeys.
ps.sendKeys('data' + ControlKey.TAB + ControlKey.TAB + 'more data' +
ControlKey.ENTER);

// Le mot-clé "yield" doit être utilisé devant les appels de fonction
"wait" et "ui".
// Il indique au navigateur de suspendre l'exécution de la macro
jusqu'au retour
// de la fonction wait (asynchrone). Se reporter à la documentation pour
connaître les fonctions
// qui nécessitent le mot-clé yield.
yield wait.forCursor(new Position(10, 26));
ps.sendKeys('accounts' + ControlKey.ENTER);

// Possibilité d'attendre que du texte s'affiche dans certaines zones de
l'écran.
yield wait.forText('ACCOUNTS', new Position(3, 36)) ;
ps.sendKeys('1' + ControlKey.ENTER);

// Toutes les fonctions wait subissent un timeout si les critères ne
sont pas respectés dans un délai.
// Possibilité d'augmenter le timeout avec un paramètre facultatif dans
les fonctions wait (en millisecondes).
// Tous les timeouts sont spécifiés en millisecondes et leur valeur par
défaut est de 10 secondes (10 000 ms).
yield wait.forCursor(new Position(1, 1), 15000);
ps.sendKeys('A' + ControlKey.ENTER);

// PS fournit la fonction getText pour lire du texte sur l'écran.
numberOfAccounts = ps.getText(new Position(12, 3), 5);

// Utiliser l'objet ui pour afficher certaines données de l'écran.
ui.message('Nombre de comptes actifs : ' + numberOfAccounts);

// L'instruction try/catch permet de capturer toutes les erreurs et de
les signaler à un emplacement centralisé.
} catch (error) {
// Nous utilisons à nouveau l'objet ui pour afficher un message
signalant qu'une erreur s'est produite.
yield ui.message('Error: ' + error.message);
}
// Fin de la macro générée
});

```

```
// Exécuter la macro et renvoyer les résultats dans l'outil d'exécution de  
macro.  
// L'instruction return est requise, car l'application s'en sert  
// pour déterminer si la macro a réussi et le moment de fin de son  
exécution.  
return macro();
```

## Interaction avec l'utilisateur

---

Cet exemple illustre comment utiliser les méthodes de l'API fournies pour demander à l'utilisateur de saisir des données ou lui envoyer un message.

```
var macro = createMacro(function*(){
  'use strict';

  // L'objet "ui" fournit des fonctions pour inviter l'utilisateur à fournir
  des informations et pour afficher des informations.

  // Déclarer des variables pour une utilisation ultérieure.
  var username;
  var password;
  var flavor;
  var scoops;

  // Début de la macro générée
  var ps = session.getPresentationSpace();

  try {
    // Inviter l'utilisateur à entrer son nom et stockez-le dans une
    variable.
    // Le mot-clé "yield" est nécessaire pour bloquer l'exécution lors de
    l'attente de l'entrée utilisateur.
    username = yield ui.prompt('Entrez votre nom d'utilisateur');

    // Inviter l'utilisateur à entrer une valeur en lui indiquant une valeur
    par défaut.
    flavor = yield ui.prompt('Quel est votre parfum de glace préféré ?',
    'Chocolat');

    // Inviter l'utilisateur à entrer des informations personnelles à l'aide
    de l'option "mask" et le champ de saisie est masqué lors de la saisie.
    // Si un paramètre n'est pas utilisé, la valeur "null" peut être utilisée
    pour indiquer qu'il ne doit pas être utilisé.
    // Nous illustrons ce principe ici en indiquant que nous n'avons pas
    besoin d'afficher une valeur par défaut.
    password = yield ui.prompt('Entrez votre mot de passe', null, true);

    // La fonction prompt renvoie la valeur null si l'utilisateur clique sur
    le bouton "Annuler" au lieu du bouton "OK".
    // Une façon de gérer cette situation est d'envelopper l'appel dans un
    bloc try/catch.
    scoops = yield ui.prompt('Combien de boules voulez-vous ?');
    if (scoops === null) {
      // Cette fonction quitte la macro.
      return;
      // Peut générer une erreur et l'intégrer dans l'objet "catch" ci-
      dessous.
    }
  }
}
```

```
// Utiliser les valeurs collectées pour commander une glace.
ps.sendKeys(username + ControlKey.TAB + password + ControlKey.ENTER);
yield wait.forCursor(new Position(5, 1));
ps.sendKeys(flavor + ControlKey.TAB + scoops + ControlKey.ENTER);

// Afficher un message à l'utilisateur. L'utilisation du mot-clé "yield"
devant l'appel bloque
// l'exécution ultérieure de la macro jusqu'à ce que l'utilisateur clique
sur le bouton "OK".
yield ui.message('Commande réussie. Bonne dégustation de vos ' + scoops +
' boules de glace ' + flavor + username + '!');
} catch (error) {
// Ici nous utilisons l'objet ui pour afficher un message signalant
qu'une erreur s'est produite.
yield ui.message(error.message);
}
// Fin de la macro générée

});

return macro();
```

## Navigation dans les données

---

Cet exemple illustre comment naviguer entre un nombre variable d'écrans et traiter les données de chaque écran.

```

// Créer une fonction de macro.
var macro = createMacro(function*(){
  'use strict';

  // Créer des variables pour une utilisation ultérieure.
  var password;
  var accountNumber;
  var transactionCount = 0;
  var row = 0;

  // Obtenir une référence à l'objet PresentationSpace.
  var ps = session.getPresentationSpace();

  try {
    // Entrer le nom d'utilisateur et le mot de passe pour se connecter à
    l'application.
    yield wait.forCursor(new Position(19, 48));
    ps.sendKeys('bjones' + ControlKey.TAB);

    yield wait.forCursor(new Position(20, 48));
    password = yield ui.prompt('Mot de passe :', null, true);
    ps.sendKeys(password);
    ps.sendKeys(ControlKey.ENTER);

    // Entrer une commande d'application.
    yield wait.forCursor(new Position(20, 38));
    ps.sendKeys('4');
    ps.sendKeys(ControlKey.ENTER);

    // Afficher les transactions pour un compte.
    yield wait.forCursor(new Position(13, 25));
    ps.sendKeys('2');
    // Entrer un numéro de compte. Codé en dur pour des raisons de
    simplicité.
    yield wait.forCursor(new Position(15, 25));
    accountNumber = yield ui.prompt('Numéro de compte :', '167439459');
    ps.sendKeys(accountNumber);
    ps.sendKeys(ControlKey.ENTER);

    // Attendre l'écran de profil de compte.
    yield wait.forText('PROFIL DE COMPTE', new Position(3, 33));

    // Rechercher du texte qui indique que la dernière page de
    l'enregistrement est atteinte.
    while (ps.getText(new Position(22, 12), 9) !== 'LAST PAGE') {

```

```

    // Tant que la dernière page de l'enregistrement n'est pas atteinte,
    accéder à la page suivante d'enregistrements.
    ps.sendKeys(ControlKey.PF2);
    yield wait.forCursor(new Position(1, 1));

    // Si la position du curseur ne change pas entre des écrans
    d'enregistrement, et qu'aucun texte
    // n'est affiché à l'écran, vous pouvez vérifier qu'un écran est mis à
    jour ou attendre
    // une période fixe après l'envoi d'une touche d'aide pour que l'écran
    se stabilise.
    // Par exemple :
    // yield wait.forFixedTime(1000);

    // Pour chacune des lignes, incrémenter la variable count si elle
    contient des données.
    for (row = 5; row <= 21; row++) {

        // Il y a 2 colonnes à l'écran. Vérifier les données de la colonne 1.
        // Dans cet exemple, nous savons que la présence d'un espace à une
        // position particulière indique qu'il y a une transaction.
        if (ps.getText(new Position(row, 8), 1) !== ' ') {
            transactionCount++;
        }
        // Vérifier les données de la colonne 2.
        if (ps.getText(new Position(row, 49), 1) !== ' ') {
            transactionCount++;
        }
    }
}

// Après avoir parcouru toutes les pages d'enregistrement, afficher le
nombre d'enregistrements dans une zone de message.
yield ui.message(' + transactionCount + ' enregistrements ont été trouvés
pour le compte ' + accountNumber + '.');

// Déconnexion de l'application.
ps.sendKeys(ControlKey.PF13);
ps.sendKeys(ControlKey.PF12);

// L'instruction try/catch permet de capturer toutes les erreurs et de
les signaler à un emplacement centralisé.
} catch (error) {
    // Nous utilisons ici l'objet ui pour afficher un message qui signale
    qu'une erreur s'est produite.
    yield ui.message(error.message);
}

```



```
});  
  
// Ici nous exécutons la macro et renvoyons les résultats dans l'outil  
d'exécution de macro.  
// L'instruction return est requise, car l'application s'en  
// sert pour déterminer si la macro a réussi.  
return macro();
```

## Appel d'un service Web

Cet exemple montre comment effectuer un appel AJAX/REST directement à partir d'une macro vers un service Web. Vous pouvez intégrer des données de votre application hôte à l'appel du service Web ou du service Web à votre application hôte.

Dans cet exemple, nous appelons le service REST Verastream Host Integrator (VHI) CICSActsDemo. Toutefois, vous pouvez facilement adapter le code pour appeler n'importe quel service Web. Vous n'êtes pas limité à VHI.

Dans l'exemple, l'appel passe par un proxy configuré sur le serveur de session (illustré ci-dessous) pour éviter une complication de type « même stratégie d'origine ». Si vous utilisez un service Web qui prend en charge le [partage CORS \(Cross-Origin Resource Sharing\)](#) et que vous utilisez un navigateur moderne, le proxy n'est pas nécessaire.

La bibliothèque jQuery étant disponible dans les macros, vous pouvez utiliser la fonction \$.post() directement pour appeler des services REST.

Cet exemple montre également comment envelopper un appel REST jQuery dans un nouvel objet Promise. L'objet Promise renvoyé par la fonction personnalisée ci-dessous permet d'utiliser le mot-clé « yield » dans le code de la macro principale. Cela permet à l'exécution de la macro principale d'attendre la fin de l'appel de service avant de continuer.

```

var macro = createMacro(function*() {
  'use strict';

  // Créer quelques variables pour une utilisation ultérieure.
  var username;
  var password;
  var accountNumber;
  var accountDetails;

  // Créer une fonction qui effectue un appel AJAX/REST au service Web VHI.
  // Possibilité de l'ajuster pour appeler n'importe quel service Web, pas
  // seulement VHI.
  // Si CORS n'est pas utilisé, la requête doit probablement transiter par
  // un proxy sur le serveur de session. Pour plus d'informations, se
  // reporter aux notes sur l'exemple.
  /**
   * Fonction d'aide codée à la main pour encapsuler les paramètres AJAX/
  REST, appeler le
   * service REST et renvoyer les résultats dans un objet Promise.
   * @param {Number} Numéro de compte à envoyer à la requête REST.
   * @param {String} Nom d'utilisateur pour accéder au service REST.
   * @param {String} Mot de passe pour accéder au service REST.
   * @return {Promise} Contient les résultats $.post() compatibles avec
  yield.
   */
  var getAccountDetails = function (acctNum, username, password)
    { var url = "proxy1/model/CICSAcctsDemo/GetAccountDetail";
      var args = {"filters": {"N° compte": acctNum}, "envVars": {"Nom
  d'utilisateur": username, "Mot de passe": password}};

      // Envelopper un appel jQuery AJAX/HTTP POST dans un nouvel objet
  Promise.
      // L'objet Promise renvoyé ici permet à la macro d'être générée/attendre
      // la fin d'exécution.
      return Promise.resolve($.post(url, JSON.stringify(args)))
        .catch(function (error) {
          // Assigner les erreurs qui se produisent dans l'appel jQuery à notre
  objet Promise.
          throw new Error('REST API Error: ' + error.statusText);
        });
    };

  // Début de la macro générée
  var ps = session.getPresentationSpace();
  try {
    // Possibilité d'interagir avec l'hôte ici, de se connecter à une

```

```

application hôte, etc.
// Collecter le nom d'utilisateur et le mot de passe
username = yield ui.prompt('Nom d'utilisateur :');
password = yield ui.prompt('Mot de passe :', null, true);
accountNumber = yield ui.prompt('Numéro de compte :');
if (!username || !password || !accountNumber) {
    throw new Error('Nom d'utilisateur ou mot de passe non spécifié');
}

// Appeler un service REST externe et générer/attendre la fin de l'appel.
accountDetails = yield getAccountDetails(accountNumber, username,
password);

// Nous disposons à présent des données de notre service externe.
// Possibilité d'intégrer les données dans notre application hôte locale
ou de les afficher pour l'utilisateur.
// Pour cet exemple, nous affichons simplement les détails du compte
correspondant.
if (accountDetails.result && accountDetails.result.length > 0) {
    yield ui.message(accountDetails.result[0].FirstName + ' $' +
accountDetails.result[0].AcctBalance);
} else {
    yield ui.message('Aucun enregistrement trouvé pour le compte : ' +
accountNumber);
}
} catch (error) {
// Si une erreur se produit pendant l'appel AJAX/REST
// ou la collecte du nom d'utilisateur/mot de passe, nous aboutissons
ici.
yield ui.message(error.message);
}
});

// Exécution de notre macro.
return macro();

```

## Prise en charge du proxy de script entre origines

Si vous avez des services Web qui ne prennent pas en charge CORS, les appels AJAX/REST échouent s'ils tentent d'accéder à un serveur autre que celui d'où provient l'application Host Access for the Cloud. Il s'agit d'une fonction de sécurité du navigateur.

Le serveur Host Access for the Cloud permet d'effectuer un proxy explicite vers les serveurs distants approuvés.

- Ouvrez `..\<install_dir>\sessionserver\microservice\sessionserver\service.yml` à des fins de modification.
- Dans la section `env`, ajoutez :

```
name : zfe.proxy.mappings
value : proxy-path=proxy-to-address
```

Où `proxy-path` correspond à l'assignation d'URL souhaitée et `proxy-to-address` à l'URL vers laquelle le proxy de l'appel est effectué.

- Dans cet exemple :

```
name : zfe.proxy.mappings
value : proxy1=http://remote-vhi-server:9680/vhi-rs/
```

Les appels effectués `<server:port>/proxy1` sont mis en proxy vers `http://remote-vhi-server:9680/vhi-rs/`.

- Il est possible de spécifier plusieurs assignations de proxy en les séparant par une virgule.
- N'oubliez pas que même lorsqu'un serveur REST prend en charge les en-têtes CORS, il se peut que certains navigateurs plus anciens ne le fassent pas. Cet exemple peut donc toujours être pertinent.

#### conseil

Votre fichier `service.yml` peut être remplacé lorsque vous redéployez Host Access for the Cloud. Effectuez dès lors toujours une sauvegarde de vos fichiers.

## Utilisation des cellules de données et des attributs

Cette macro indique comment utiliser DataCells et AttributeSet pour inspecter le texte et les attributs d'une ligne/colonne donnée à l'écran. Dans cet exemple, vous pouvez constater :

- Comment obtenir une collection de DataCells pour une position et une longueur données.
- Comment itérer DataCells pour créer une chaîne de texte.
- Comment vous pouvez également effectuer une opération similaire à l'aide de `getText()` à des fins de comparaison.
- Enfin, comment utiliser des attributs, obtenir une liste de chaînes ou déterminer si des attributs spécifiques sont définis à un emplacement d'écran donné.

```

var macro = createMacro(function*() {
  'use strict';

  // Obtenir l'objet PresentationSpace pour interagir avec l'hôte.
  var ps = session.getPresentationSpace();

  // Déclarer des variables pour une utilisateur ultérieure.
  var cells;
  var text;
  var attrs;

  // Définir le timeout par défaut pour les fonctions "wait".
  wait.setDefaultTimeout(10000);

  // Exemple de macro pour utiliser des objets DataCells et Attributes
  try {
    yield wait.forCursor(new Position(24, 2));

    // Obtenir des objets DataCells pour l'espace de présentation.
    // La ligne 19, colonne 3 est l'invite, longueur de 35 caractères.
    // "Choisissez l'une des commandes suivantes :"
    cells = ps.getDataCells({row:19, col:3}, 35);
    text = '';

    // Vous pouvez afficher du texte avec getText.
    yield ui.message("Texte d'écran : " + ps.getText({row:19, col:3}, 35));

    // Vous pouvez aussi assembler le texte de DataCells à chaque position.
    for(var index = 0; index < cells.length; index++) {
      text = text.concat(cells[index].getChar());
    }
    // Et afficher le texte.
    yield ui.message("Texte des cellules : " + text);

    // Obtenir les attributs pour le premier objet DataCell. (cell[0])
    attrs = cells[0].getAttributes();

    // Afficher si nous avons des attributs dans la cellule de données.
    yield ui.message("Jeu d'attributs vide : " + attrs.isEmpty());

    // Afficher le nombre d'attributs définis.
    yield ui.message("Nombre d'attributs : " + attrs.size());

    // Afficher les attributs définis.
    yield ui.message("Attributs : " + attrs.toString());
  }
});

```

```

// Afficher à présent si l'attribut à haute intensité est défini.
yield ui.message("Haute intensité : "
                + attrs.contains(Attribute.HIGH_INTENSITY));

// Afficher à présent si l'attribut souligné est défini.
yield ui.message("Souligné : "
                + attrs.contains(Attribute.UNDERLINE));

// Afficher à présent si des attributs alphanumériques, intensifiés et
détectables par stylet sont définis.
yield ui.message("Alphanumérique, intensifié et détectable par stylet : "
+
                attrs.containsAll([Attribute.ALPHA_NUMERIC,
Attribute.HIGH_INTENSITY, Attribute.PEN_DETECTABLE]));

// Afficher à présent si des attributs soulignés, intensifiés et
détectables par stylet sont définis.
yield ui.message("Souligné, intensifié et détectable par stylet : " +
                attrs.containsAll([Attribute.UNDERLINE,
Attribute.HIGH_INTENSITY, Attribute.PEN_DETECTABLE]));
} catch (error) {
    yield ui.message(error);
}
// Fin de la macro générée
});

// Exécuter la macro.
return macro();

```

## Utilisation de champs et de listes de champs

Cet exemple de macro illustre comment utiliser des fonctions courantes pour interagir avec les champs de l'API Macro. Par exemple, comment obtenir le texte d'un champ, afficher les informations d'un champ et utiliser `field.setText` comme solution alternative à `sendKeys` pour interagir avec l'hôte.

### remarque

En raison de considérations relatives au navigateur, `ui.message` réduit les chaînes d'espaces en un seul espace. Les espaces sont conservés dans le code JavaScript réel.

```

var macro = createMacro(function*() {
    'use strict';

    // Obtenir l'objet PresentationSpace pour interagir avec l'hôte.
    var ps = session.getPresentationSpace();

    // Déclarer des variables pour une utilisation ultérieure.
    var fields;
    var field;
    var searchString = 'z/VM';

    // Définir le timeout par défaut pour les fonctions "wait".
    wait.setDefaultTimeout(10000);

    // Exemple de macro pour utiliser FieldList et Fields.
    try {
        yield wait.forCursor(new Position(24, 2));

        // Obtenir la liste de champs.
        fields = ps.getFields();

        // Parcourir toute la liste de champs et afficher les informations des
        champs.
        for(var index = 0; index < fields.size(); index++) {
            field = fields.get(index);

            yield ui.message("Champ " + index + " info : " + field.toString());
        }

        yield ui.message("Rechercher à présent un champ contenant le texte '" +
searchString + "'");
        field = fields.findField(new Position(1, 1), searchString);

        if(field !== null) {
            yield ui.message("Informations du champ trouvé : " +
field.toString());
            yield ui.message("Le premier plan du champ trouvé est vert ? " +
(Color.GREEN === field.getForegroundColor()));
            yield ui.message("L'arrière-plan du champ trouvé est par défaut ? " +
(Color.BLANK_UNSPECIFIED === field.getBackgroundColor()));
        }

        // Rechercher à présent un champ de commande et le modifier.
        field = fields.findField(new Position(23, 80));
        if(field !== null) {
            field.setText("cics");
        }
    }
});

```

```

}

yield ui.message("Cliquez pour envoyer 'cics' à l'hôte.");
ps.sendKeys(ControlKey.ENTER);

// Attendre un nouvel écran, obtenir de nouveaux champs.
yield wait.forCursor(new Position(10, 26));
fields = ps.getFields();

// Rechercher un champ d'utilisateur et le définir.
field = fields.findField(new Position(10, 24));
if(field != null) {
    field.setText("nom_utilisateur");
}

// Rechercher un champ de mot de passe et le définir.
field = fields.findField(new Position(11, 24));
if(field != null) {
    field.setText("mot_passe");
}

yield ui.message("Cliquez pour envoyer la connexion à l'hôte.");
ps.sendKeys(ControlKey.ENTER);

// Attendre un nouvel écran, obtenir de nouveaux champs.
yield wait.forCursor(new Position(1, 1));
fields = ps.getFields();

// Rechercher un champ de commande et définir la commande de déconnexion.
field = fields.findField(new Position(24, 45));
if(field != null) {
    field.setText("cesf logoff");
}

yield ui.message("Cliquez pour envoyer la déconnexion à l'hôte.");
ps.sendKeys(ControlKey.ENTER);

} catch (error) {
    yield ui.message(error);
}
// Fin de la macro générée
});

// Exécuter la macro.
return macro();

```



## Macro d'ouverture de session automatique pour les mainframes

---

Dans cet exemple, l'objet Autosignon sert à créer une macro qui utilise les informations d'identification associées à un utilisateur pour obtenir un ticket d'accès à partir du serveur DCAS (Digital Certificate Access Server).

```

var macro = createMacro(function*() {
  'use strict';

  // Obtenir l'objet PresentationSpace pour interagir avec l'hôte.
  var ps = session.getPresentationSpace();

  // Variable pour le ticket d'accès de connexion.
  var passTicket;

  // ID d'application de connexion.
  var appId = 'CICSV41A';

  // Définir le timeout par défaut pour les fonctions "wait".
  wait.setDefaultTimeout(10000);

  // Début de la macro générée
  try {
    yield wait.forCursor(new Position(24, 2));

    // Obtenir un ticket d'accès depuis DCAS.
    passTicket = yield autoSignon.getPassTicket(appId);

    ps.sendKeys('cics');
    ps.sendKeys(ControlKey.ENTER);

    yield wait.forCursor(new Position(10, 26));

    // Remplacer le nom d'utilisateur généré par sendUserName(passTicket) ...
    yield autoSignon.sendUserName(passTicket);

    // ps.sendKeys('bvtst01' + ControlKey.TAB + ControlKey.TAB);
    ps.sendKeys(ControlKey.TAB + ControlKey.TAB);

    yield wait.forCursor(new Position(11, 26));

    // Remplacer le mot de passe généré par sendPassword(passTicket) ...
    yield autoSignon.sendPassword(passTicket);

    // var userInput3 = yield ui.prompt('Mot de passe :', '', true);
    // if (userInput3 === null) {
    //   // throw new Error('Mot de passe non fourni');
    // }
    // ps.sendKeys(userInput3);
    ps.sendKeys(ControlKey.ENTER);

    yield wait.forCursor(new Position(1, 1));
  }
});

```

```
yield ui.message('Connecté. Se déconnecter. ');
ps.sendKeys('cesf logoff');
ps.sendKeys(ControlKey.ENTER);
} catch (error) {
yield ui.message(error);
}
// Fin de la macro générée
});

// Exécuter la macro.
return macro();
```

## Utilisation du transfert de fichiers (IND\$File)

Cette série d'exemples de macros montre comment utiliser l'API de transfert de fichiers pour récupérer une liste de fichiers, télécharger un fichier et charger un fichier vers un hôte 3270.

### remarque

Vous devez être connecté et être à une invite avant d'exécuter ces macros.

[Répertoire des fichiers](#)

[Télécharger un fichier](#)

[Charger un fichier](#)

---

### Répertoire des fichiers

Cette macro explique comment utiliser l'API de transfert de fichiers pour récupérer une liste de fichiers sur un hôte 3270 à l'aide du transfert IND\$File. L'objet de transfert IND\$File est récupéré de l'usine de transfert de fichiers, puis utilisé pour obtenir un tableau d'objets HostFile depuis TSO ou CMS.

```

var macro = createMacro(function*() {
  'use strict';

  try {
    var fileTransfer = fileTransferFactory.getInd$File();
    var hostFiles = yield fileTransfer.getHostFileListing();

    yield ui.message(' + hostFiles.length + ' fichiers trouvés');
    if (hostFiles.length > 0) {
      var firstFile = hostFiles[0];
      var msg1 = 'Le nom du catalogue est ' + firstFile.getParent() + '. ';
      var msg2 = 'Le premier fichier est ' + firstFile.getName();
      yield ui.message(msg1 + msg2);
    }
  } catch (error) {
    yield ui.message(error);
  }
});

// Exécuter la macro.
return macro();

```

## Télécharger un fichier

Cette macro indique comment utiliser l'API de transfert de fichiers pour télécharger un fichier à partir d'un hôte 3270 à l'aide du transfert IND\$File. L'objet de transfert IND\$File est récupéré de l'usine de transfert de fichiers. Dans cet exemple, la méthode de transfert est définie sur ASCII pour démontrer l'utilisation de la fonction setTransferOptions.

L'exemple de macro télécharge le premier fichier renvoyé par un appel à getHostFileListing en créant un URI de téléchargement avec un appel à la fonction getDownloadUrl. La macro peut être utilisée dans un environnement CMS ou TSO, mais l'option doit être spécifiée sur la première ligne ou le code modifié légèrement pour le système prévu.

```

var hostEnvironment = 'CMS'; // 'TSO'
// Construire le chemin de fichier, catalogue/nom.fichier ou catalogue/
partition/fichier
function getPath (fileNode) {
    var prefix = fileNode.getParent() ? fileNode.getParent() + '/' : '';
    return prefix + fileNode.getName();
}

var macro = createMacro(function*() {
    'use strict';

    try {
        var fileTransfer = fileTransferFactory.getInd$File();

        // Les options transferMethod sont 'binary' et 'ascii'.
        fileTransfer.setTransferOptions({transferMethod: 'ascii'});

        // Cette démo récupère le premier fichier renvoyé dans la liste.
        var hostFiles = yield fileTransfer.getHostFileListing();
        var firstHostFile = hostFiles[0];

        if (hostEnvironment === 'CMS') {
            yield wait.forText('Prêt', new Position(1,1), 5000);
        }

        // Télécharger.
        // Si vous connaissez déjà le chemin du fichier souhaité, transmettez-le
à getDownloadURL().
        var downloadUrl = fileTransfer.getDownloadURL(getPath(firstHostFile));

        // Modifie l'emplacement du navigateur. Les résultats peuvent varier en
fonction du navigateur utilisé.
        window.location = downloadUrl;

        // Si vous souhaitez lire le contenu du fichier dans une variable au lieu
de le
// télécharger, vous pouvez utiliser jQuery
// var fileContents = yield $.get(downloadUrl);

    } catch (error) {
        yield ui.message(error);
    }
});

// Exécuter la macro.
return macro();

```