Orbix 3.0.1- Patch 20

Release Notes

March 2000

Contents

Introduction

This document contains information about Orbix 3.0.1, including build information, information about new features, and details of faults that have been fixed in this release.

This document is divided into five main sections, each corresponding to one of the components of Orbix 3.0:

- Orbix 3.0.1-Patch 20
- Orbix Code Generation Toolkit 3.0.1
- OrbixCOMet 3.0.1-Patch 20
- OrbixNames 3.0.1-Patch 20
- Orbix Wonderwall 3.0.1

Orbix 3.0.1-Patch 20

This section describes changes made in Orbix 3.0.1-Patch 20

Development Environments

This section describes the compiler and operating system versions that Orbix 3.0.1-Patch 20 has been built and tested with. The following applies to both multi-threaded and single-threaded variants of Orbix 3.0.1-Patch 20.

Solaris 2.5.1

Orbix 3.0.1-Patch 20 has been built on Solaris 2.5.1 using the SPARC C++ compiler version 4.2. Patches 103640-27 (kernel and `libthread` patch) and 103627-10 (linker patch) should be installed on your system. Both patches are included in the "Solaris 2.5.1 Recommended Patch Cluster" from Sun Microsystems, dated May 26.

For Solaris 2.6, patch 105568-11 or higher (`libthread.so.1` patch) should be installed on your system. This patch has a dependency on patch 105210 (`libc.so.1` patch), that is patch 105210 must be installed.

Testing of Orbix 3.0.1-Patch 20 was carried out on Solaris 2.5.1 and 2.6 and SPARC compiler version 4.2.

The Orbix 3.0.1-Patch 20 libraries contain RTTI information.

HP-UX 10.20

Orbix 3.0.1 has been built and tested on HP-UX10.20 using the ANSI compiler version A.10.36 with patch phss-13124 installed.

Testing of Orbix 3.0.1 was carried out on HP-UX10.20 on 700 and 800 series machines. Orbix has been compiled with `+DAportable`.

Orbix must now be compiled with the `-ext` flag to enable support for the `long long` and `unsigned long long` data types. Failure to do this results in compilation errors.

HP-UX 11.00

Orbix 3.0.1 has been built and tested on HP-UX11.00 using the ANSI compiler version A.03.13.

NT 4.0

Orbix 3.0.1 has been built and tested on NT 4 with service pack 4 installed, using the VC compiler version 6 service pack 2.

Interoperability

This section describes the interoperability between Orbix 3.0.1-Patch 20 and other IONA products.

Interoperability between Orbix 3.0.1-Patch 20 and Orbix 2000

Orbix 3.0.1- Patch 20 has been tested for interoperability against the following IONA products:

- Orbix 2000

- OrbixWeb 3.2

The *Interoperability Guide* describes how to configure applications that use a mixture of IONA products, and any feature limitations that apply to such interoperating systems. The *Interoperability Guide* can be located at:

```
http://www.iona.com/docs/orbix2000/orbix200010.html
```

Note: Configuration changes are required to enable an Orbix 2000 client to invoke an Orbix 3 server. For more information, see the article "Launch and Invoke Rights of Orbix 3 Servers" in the *Interoperability Guide.*

Interoperability between Orbix 3.0.1-Patch 20 and the OrbixOTM, OrbixTalk and OrbixNotification Products

Please note that users of OrbixOTM and users of the IONA Messaging Products, OrbixTalk and OrbixNotification should NOT upgrade to this patch level. Enhancements introduced in the Orbix patch do affect binary compatibility with the Messaging and OrbixOTM products.

For OrbixTalk users, new patches will be made available alongside the latest Orbix patches.

All users of OrbixNotification and OrbixOTM should wait until the next minor releases of these products become available.

Please contact Customer Services for release dates and information on IONA product and patch releases.

Interoperability between Orbix 3.0.1-Patch 20, OrbixNames 3.0.1 and OrbixWeb 3.2-Patch 5

The `IT_DEFAULT_CLASSPATH` configuration variable must be altered to include `opt/iona/lib/OrbixNamesUtils.jar`. This is done in the `common.cfg` configuration file.

Due to an OrbixWeb issue, you must add `orbixweb3.jar` to the configuration file:

```
OrbixWeb {
        IT_NAMES_SERVER_HOST="hostname";
        IT_NS_PORT="1570";
};
```

If you do not already have an `orbixweb3.cfg` configuration file, note that this file should be included in the `iona.cfg` file.

Due to a change in the format for the Names Repository you must clean out the repository and repopulate it with the new OrbixNames server. To do this, go to the directory where the Names Repository is (default is `opt/iona/config/Repositories/namesRep/`) and delete all files (using `rm-f`) and then restart the Naming Service.

Names GUI: For interoperability with the Orbix 2000 Naming Service you will must add a line to the `NamesBrowser_en_US.properties` file (located in the `bin` directory) with the `IOR` of the Naming Service under the variable `ARTNS`. For example, `ARTNS=IOR:00…`

When you start the Names Browser and try to connect to a Naming Service, there is a checkbox that gives you the option to connect to this Naming Service.

For further information on OrbixWeb 3.2- Patch 5 see:

```
http://www.iona.com/docs/orbixweb.html
```

Bugs Fixed in Orbix 3.0.1-Patch 20

This section describes the bugs fixed in this release. All bugs are cross platform unless otherwise stated. The bugs are broken down by module and described in terms of the following:

- **Bug Number**

  This is the reference number used by the development teams to track bugs, which may in turn relate to one or more PRs (problem reports) as reported by customers.

- **Synopsis**

  This is a short description of the reported problem. A description of the fix is included where necessary.

Bug Number	Synopsis
22440	Foreign IORs are mishandled by Orbix 3.0.1
36260	Server crash if method raises user exception when passed obj ref is deleted already.
43160	White space in idl macros causes problems.
48200	Cannot have OrbixTalk 3.0 talker and listener in the same process space with the same topic name.
48860	Calling `_non_existent()` launches an exited server instead of checking it.
51585	Empty `#defines` should not generate errors in Orbix.
51939	`_bind()` problem between HPUX 11 and other OS when rebinding after killing the daemon.
52018	Any within sequence of struct within a struct performs badly in Orbix 3.0.
52069	Purify FNH error caused by `BOA_init()`.
52284	The `bidirectionalIIOP(1)` call used with OrbixNames 3.0 causes exception 10100.
52428	Multithreaded problem with file descriptor blocking.
52515	Orbix 3 server core dumps when an OrbixWeb client passes a union into an Any.
52599	Passing and returning Anys that contain a simple struct gives Purify UMR errors.
52664	`_hash` does not return proper id for object reference directly after it is created on server-side.
52683	The IOR's type id is wrong for Orbix 3.0.1 with the `grid_dsi` demo when an `Object_to_String` is performed.
52778	OrbixNames utilities appear to ignore the -h flag.
52847	Problem with Orbix 3.0.1 Patch 1 Any encoding/decoding.
52901	Allocbuf in Solaris 3.0 does not return null when memory can not be allocated.
52951	NT pid reuse problem.
52963	Error 10170 encoding/decoding error when transmitting an Any if the server was compiled without the skeleton code.
52987	TypeCode flag does not work as expected when struct is passed through an Any.

53041	FMMs within orbixlibs, when running grid demo through Purify.
53082	DII problem with an operation returning a sequence of Contained object.
53097	Orbix 3.0.1 MT client/server deadlock.
53216	Cannot have OrbixTalk 3.0 talker and listener in the same process space with the same topic name.
53295	DNS lookup caused by gethostbyaddr results in slow bind.
53300	Orbix 3.0.1 server has memory leaks when passed a struct to an Any as an IN parameter. Note that the struct contains a sequence of string.
53381	Memory leaks in Orbix 3.0 on WinNT.
53469	Object Reference count is doubled when Objects are passed in with a sequence of Anys.
53548	struct data in CORBA::Any thrown in an exception are not being transferred between Orbix s1190-2.3.4 server and Orbix s1475-3.0 client.
53647	Orbix cdr coder makes IIOP profile encapsularion the wrong length, it adds 4 nulls at the end.
53673	IDL const values, defined by nested typedefs, cannot be retrieved from the interface repository.
53686	Unions with signed integer types as discriminant and const as the case are broken with the IFR.
53923	The `_non_existent()` call from a Visibroker 3.3 client to an Orbix 3.0.1 server does not work.
53955	Problem marshalling doubles - 8 byte alignment when buffer arrives on client side is not guaranteed.
54090	Unbounded sequences with maximums and a length less than maximum cause problems coming back from a server under POOP.
54093	Orbix3.0.1 server (on Solaris) closes connection to the client after receiving an incoming request of a certain type.
54106	`addForeignFD()` and `isForeignFD()` do not give expected results on NT Orbix 3.0.1.
54255	Putting all your API, method calls, attributes, and operations into the one try catch block in 3.0.1 will generate an errorCode for NT.
54432	Throwing system exception from ServiceContextHandler causes core dump when IDL method contains a raises clause.

Known Problems, Workarounds, and Tips

Registering Orbix 2000 servers that use modules with Orbix Naming Service does work using the C++ naming service utilities (putns). However, Orbix 3 clients can only use the object references obtained from the Naming Service in this case. Orbix 2000 clients cannot use these object references. Servers that implement IDL without modules do not have this problem. This is bug #54641.

OrbixCOMet Desktop 3.0.1-Patch 20

This section describes OrbixCOMet Desktop 3.0.1-Patch 20

Installation

To install, run "`install.exe`" or "`setup.exe`". This will copy all necessary files and register the appropriate DLLs etc. In the event of a crash at the end of the install i.e. when registering the DLLs, simply run the "`regcomet.bat`" batch file to ensure that everything has been registered correctly.

If the install program crashes during the DLL registration or reports that various DLL's cannot be found, you can manually register the OrbixCOMet runtime using the batch file `regCOMet.bat` located in `\iona\COMet\bin`.

Licensing Issues

OrbixCOMet Desktop 3.0.1-Patch 20 requires a valid license to function correctly. During installation you will be given the opportunity to enter a license key; if you choose not to do so, you will be granted an evaluation license which is valid for 21 days from the date of installation.

During that time you will be notified periodically that you are using an evaluation copy. If you get such a reminder, simply re-run your application to continue.

The messages below are related to licensing issues, and may appear under various circumstances:

- **Message**

  `"COMet Licensing error: Invalid License Key format"`
  `"COMet Licensing error: Missing License Key"`

  Meaning

  The license key has been corrupted/deleted.

  Action

  Reinstall OrbixCOMet.

- **Message**

  `"COMet Licensing error: License Key has expired"`

  Meaning

  The product has deactivated itself until a valid, up-to-date license has been obtained from IONA.

  Action

  Contact `sales@iona.com`

- **Message**

  ```
  "COMet Licensing error: Missing License DLL \
  (Have you registered ITLicense.DLL?)"
  ```

  Meaning

  The OrbixCOMet licensing server could not be found.

  Action

  Ensure that ITLicense.dll and it_licps.dll are both registered, using
  regsvr32.exe.

  ```
  regsvr32 ITLicense.dll
  regsvr32 it_licps.dll
  ```

If any of these errors occur, try re-licensing the product with a valid license
obtained from IONA, or re-registering the two license DLLs as described above.

- **Message**

  ```
  "COMet Eval-License Reminder : .... "
  ```

  Meaning

  This message will appear approximately every 50 runs of an OrbixCOMet
  application and provides information of how to purchase a fully licensed
  version of OrbixCOMet if you so desire.

If you subsequently receive a full license for OrbixCOMet from IONA, it should be
entered into the `orbixcomet3.0.cfg` file, under `"COMet.Licensing.IT_KEY"`.

Introduction to COMet Multimedia Presentation

To run the OrbixCOMet multimedia presentation run `\docs\cometint.exe`

Development Environments

This section describes the compiler and operating system versions that
OrbixCOMet 3.0.1-Patch 20 has been built and tested with.

Win 32

OrbixCOMet Desktop 3.0.1-Patch 20 is built and tested on NT 4.0, service pack
4, using Microsoft Visual C++ 6.0, with Visual Studio Service Pack 2 applied.
OrbixCOMet Desktop is not supported on earlier versions of NT than 4.0,
since DCOM is not available on earlier versions of NT.

OrbixCOMet Desktop 3.0.1-Patch 20 has been tested with Automation client
applications built with the following:

- PowerSoft PowerBuilder Version 6.0

- Borland Delphi 3/4

When using Delphi 4, Inprise recommend that you will make a call to `Application.Initialise();` before making any COM calls. This includes any calls to OrbixCOMet.

- Microsoft Visual Basic Version 6.0

- Microsoft Visual C++ 6.0 (SP2)

- Microsoft Excel97

- Microsoft Internet Explorer 4.0 or higher with VBScript

OrbixCOMet Desktop 3.0.1-Patch 20 has been tested with COM client applications built with the following:

- Microsoft Visual C++ 6.0 (SP2), MIDL Compiler Version 5.01.0164

OrbixCOMet Desktop 3.0.1-Patch 20 has been tested with CORBA server applications built with the following:

- Orbix 3.0.1-Patch 20

- Orbix 2000

OrbixCOMet Desktop 3.0.1-Patch 2 has been tested with CORBA client applications built with the following:

- Orbix 3.0.1-Patch 20

- Orbix 2000

Supported Mappings

1. bi-directional Automation/CORBA as per COM/CORBA Interworking Specification, OMG Document ORBOS/98-02-01, (February 01 1998)

2. bi-directional COM/CORBA as per COM/CORBA Interworking Specification, OMG Document ORBOS/98-02-01, (February 01 1998)

Usage Models

Automation

1. In-process dispatch.

2. Out-of-process dispatch.

   a) Local machine - IIOP on the wire.

   b) Remote machine - DCOM on the wire.

3. In-process dual interface.

4. Out-of-process dual interface (local/remote machine).

   a) Local machine - IIOP on the wire.

b) Remote machine - DCOM on the wire.

COM

1. In-process COM custom interfaces.

2. Out-of-process COM custom interfaces (local/remote machine).

   a) Local machine - IIOP on the wire.

   b) Remote machine - DCOM on the wire.

OrbixCOMet Desktop is a bi-directional dynamic bridge. It supports:

- COM/Automation clients of CORBA servers

- Callbacks (invocation from a CORBA server upon a COM/Automation client).

- Implementing CORBA servers in Visual Basic, PowerBuilder etc. using the IT_ServerAPI interface. For an example of how to do this, refer to the sample application in the `<COMET ROOT>\demo\vb6\bankSrv` directory.

- CORBA clients of native DCOM servers (e.g. MS Excel, MS Word etc.). For examples of this refer to the sample applications in the `<COMET ROOT>\demo\corbaclient directory`.

New Features in this Release

The documentation for OrbixCOMet 3.0.1-Patch 20 consists of:

1. GettingStarted30.pdf — Entry level information in Adobe Acrobat format.

2. ProgrammersGuide30.pdf — Complete documentation in Adobe Acrobat format.

3. COMetint.exeMulti-media — OrbixCOMet presentation

Minor Changes

 OrbixCOMet 3.0.1-Patch 20 is a release aimed purely at providing interoperability with Orbix 2000.

Other minor changes in operation with respect to earlier releases are detailed below:

- When registering `custsur.exe` as a CORBA server, the minimum recommended timeout value that should be used is 500 msecs.

- In `CORBA->DCOM` mode, when Anys containing complex types are passed as parameters from the client to the server, ensure that any relevant types are registered in the typestore by using:

      typeman -u -er <typename>

- In `CORBA->DCOM` mode, anonymous binds to CORBA wrappers have been deprecated. Instead, `ts2idl` generates a `const` string of the form:

      #ifndef _IT_COMET_ANON_

```
#define _IT_COMET_ANON_
const string IT_ANON = "IT_COMET_ANON";
#endif
```

Markers used in calls to `_bind()` should begin with this string. For example, valid markers would be:

```
"IT_COMET_ANON"
"IT_COMET_ANON1"
"IT_COMET_ANON_excelObj"
```

and so on.

As a result of this change, the default value for the `COMet.Mapping.EXTRA_REF_CORBAVIEW` configuration value is now "no", in contrast to previous 3.x releases.

For backwards compatibility, anonymous binds are allowed if the below configuration value is set to "yes" (either programmatically or within the configuration file) as shown below, but this is not recommended in most cases, use of `(D)IOrbixServerAPI` being a possible exception.

```
COMet.Mapping.ALLOW_ANON_MARKERS = "yes";
```

- A callback demo between a CORBA client and a VB server as been added - see `demo\corbaclient\callback`. This includes usage of both simple types and complex types from CORBA client to the VB server and vice-versa. It also includes an example of how to programmatically set configuration values when using OrbixCOMet's `custsur.exe` as a CORBA server.

Building/Running Demos

Runtime libraries for PowerBuilder is not included with OrbixCOMet. You will need this runtime installed if you wish to run these demos. Furthermore, in order to build the C++ CORBA servers in `<COMet Install>\demo\corbasrv`, a valid installation of Orbix3.0.1-Patch 20 is required. If you have existing CORBA servers for some of these, for example, `grid, idl_demo` which are standard Orbix demos shipped on all platforms, you may use those.

To build the C++ COM client demos you need Microsoft Visual C++ 6.0, or compatible C++ compiler.

Note that the makefiles for the CORBA servers call `putidl` to insert the IDL into the IFR. They also call `putit` to register the server in the Orbix implementation repository.

Note: C++ COM applications should not be compiled with the `/Og` or the `/Ox` switch (which implies the /Og switch). Instead, use `/Oityb1` `/Gs` for release builds. Refer to the COM demo makefiles in `<COMet Install>\demos\com` for more details. (This is due to a bug in the code optimizer in the Visual C++ compiler)

Standalone Server Support

Through the `DIOrbixServerAPI` interface OrbixCOMet allows developers to implement CORBA server objects using languages like Visual Basic, PowerScript etc. An example of how this may be used to write a CORBA server called "bank" in Visual Basic is shown below (code taken from the `<COMET ROOT>\demo\vb6\bankSrv` demo):

```
Dim orb As Object
Set orb = CreateObject("CORBA.ORB.2")
Set serverAPI = orb.GetServerAPI
Set orb = Nothing
 ' bankObj created earlier (not shown) and is our
 ' implementation object
Call serverAPI.SetObjectImpl("bank", "", bankObj)
Call serverAPI.Activate("bank")
```

The call to "Activate" above calls `impl_is_ready()` internally, to signify the server's availability to the network. As a result, an Orbix daemon is required on the machine where this application runs. This may not always be the case, and OrbixCOMet provides support for writing servers that may run without an Orbix daemon. An example of one such server is shown below (code taken from the `<COMET ROOT>\demo\vb6\standAlone` demo):

```
Dim orb As Object
Set orb = CreateObject("CORBA.ORB.2")
Set serverAPI = orb.GetServerAPI
Set orb = Nothing
 ' SetObjectImplPersistent(interface, marker, server,
 object, filename)
Call serverAPI.SetObjectImplPersistent("bank", "",
"bank", _bankObj, "c:\temp\bank.ior")
Call serverAPI.ActivatePersistent
```

The call to `SetObjectImplPersistent` specifies, in addition to the `interface/marker/server name`, a file to where the IOR for the object should be written. Prospective clients should then call `CORBA::ORB::string_to_object()` on the IOR (of course if you are using the CORBA Factory, you can take advantage of the built in support for IORs in the GetObject call c.f. `<COMET ROOT>\demo\vb6\standAlone\vbClient` for an example of this). A server written in this manner can be started persistently, without the need for a daemon on the local machine.

Important change in OrbixCOMet 3.x: As of OrbixCOMet 3.0, servers that use the `DIOrbixServerAPI` must dispatch their own Orbix events i.e. call `serverAPI.dispatchEvents` from within a timer. For example, a Visual Basic timer. Such applications must also set the `COMet.Config.AUTO_EVENTS` config value to "no".

For example:

```
Set orb = CreateObject("CORBA.ORB.2")
orb.SetConfigValue "COMet.Config.AUTO_EVENTS", "no"
Timer1.Enabled = False
// use server API
Set serverAPI = orb.GetServerAPI
Set orb = Nothing
Call serverAPI.SetObjectImpl("bank", "", bankObj)
Call serverAPI.Activate("bank")
Timer1.Interval = 20
Timer1.Enabled = True
```

And the timer function should look something like:

```
Private Sub Timer1_Timer()
Timer1.Enabled = False
serverAPI.DispatchEvents
Timer1.Enabled = True
End Sub
```

Failure to follow this approach may result in marshalling errors (such behavior was noticeable in OrbixCOMet 1.0 UR2).

Standalone IFR

The Orbix IFR may now be run as a standalone server i.e. n a configuration without a running Orbix Daemon. Essentially what was COMetIFR.exe in previous releases of OrbixCOMet has now been merged with the Orbix IFR.

To use:

When running the IFR as a standalone CORBA server, you need to tell it which port to listen on. It is recommended that this be done from a DOS command prompt.

Set the environment variable `IT_SERVER_PORT` to a port number which will be used by `IFR.exe` when saving it's IOR. (Must be an env var.)

Set OrbixCOMet configuration value `COMet.TypeMan.TYPEMAN_IFR_IOR_FILENAME` to the name of a file in which you want to store OrbixCOMetIFR's IOR. This setting is used by the IFR utilities and the OrbixCOMet utilities to retrieve the IOR.

For example: This can be in a batch file...(see `\bin\startcometifr.bat`)

```
Set IT_SERVER_PORT=2334
rem  -n   : run as persistent server
rem  -O   : Output the IOR
rem  -t 1 : Timeout after 1 sec
IFR -n -O -t 1
start IFR.exe -n
```

When using the IFR as a standalone server, the IFR utilities (`readifr.exe`, `putidl.exe` and `rmidl.exe`) should all be used with the -n switch.

Note: If you have the `TYPEMAN_IFR_IOR_FILENAME` entry set in your configuration file, you cannot have the IFR auto-launched by the daemon as well. You should use one approach or the other on any single machine.

Bugs Fixed in this Release

This release is aimed purely at providing interoperability with Orbix 2000. This release incorporates all fixes made up to and including OrbixCOMet 3.0.1- 2.

This section describes the incidents cleared in this release. All incidents are cross platform unless otherwise stated. The incidents are described in terms of **Bug Number** and **Synopsis**, as described on page 4.

Bug Number	Synopsis
30980	If a user exception occurs and you do not use inline exception handling, the error trapped is incorrect.
51904	Error in exception handling sample code.
52027	Link to release notes is incorrect.
52037	Problem with unions that have enumerated types as fields—`ts2idl` generates wrong MIDL.
52103	Narrowing a derived object using `Idispatch()`.
52178	`Typeman` crashes when priming type store with `MSOUTL8.OLB` (Microsoft Outlook express).
52179	Problem with unions with enum discriminator.
52180	Warning messages from `typeman` when priming with some type libs.
52194	Crash on creating/returning sequence of structures containing sequences of structures.

Known Problems, Workarounds and Tips

This section summarizes known issues and tips related to the OrbixCOMet 3.0.1-Patch 20 release.

The following are known issues:

- Marshalling interface pointers across Apartment boundaries when using the bridge-in process is not supported. Out of process is fine.

  This is only relevant if the Bridge objects are instantiated in a COM Single Threaded Apartment. Using OrbixCOMet objects in a Free Threaded Apartment is fine.

  It is recommended that when using OrbixCOMet in C++ you create a Multithreaded Apartment:

  ```
  CoInitializeEx(0, COINIT_MULTITHREADED);
  ```

- There is a problem with Visual Basic keeping DLL's loaded in memory even after the application has terminated. This causes OrbixCOMet to prematurely execute its shutdown procedures in response to a positive result to `CoFreeUnusedLibraries()`.

  This results in an application crash the next time the application is executed in the VB environment.

  The workaround to this problem is to programmatically set the OrbixCOMet configuration setting `COMET_SHUTDOWN_POLICY` to atexit.

  Certain versions of `regsvr32` have been known to crash when registering a handler DLL. If this behavior is seen, use the OrbixCOMet `oleregit.exe` tool instead, located in the `<COMET ROOT>\bin` directory.

For example:

To register `foo.dll`, use `oleregit foo.dll /REGSERVER`
To unregister `foo.dll`, use `oleregit foo.dll/UNREGSERVER`

When uninstalling OrbixCOMet, you may need to unregister OrbixCOMet DLLs from the OLE registry by running the unregCOmet.bat batch file located in the `COMet\bin` directory.

- When using bounded sequence from a COM client which has OrbixCOMet loaded in-process, it is recommended that any unused elements in the sequence be memset to zero '0'. OrbixCOMet will attempt to skip these unused elements, but you may get a Marshalling error if the element types are complex.

- Anys are not supported in COM. i.e. use of `ICORBA_Any`

Further Information

Note that support for OrbixCOMet is provided in the form of a knowledge base, which may be found at:

`http://www.iona.com/online/support/kb/OrbixCOMet/index.html`

A separate support contract that entitles you to email-based support queries may also be purchased. Contact `sales@iona.com` for details.

OrbixNames 3.0.1

This section describes changes in OrbixNames 3.0.1.

Development Environments

Development environment information for OrbixNames 3.0.1 is the same as that described for Orbix 3.0.1 on page 2.

Known Issues

OrbixNames 3.0.1-Patch20 release notes additions:

- You must clean out the repository and repopulate it with the new OrbixNames server. This is due to a change in the format for the Names Repository.

  To do this:

  1. Go to the directory where the Names Repository is (default is `/opt/iona/config/Repositories/NamesRep/`)

  2. Delete all files (using `rm -f`).

  3. Restart the Naming Service.

- **Names GUI:** For Interoperability with the Orbix2000 Naming Service you must add a line to `NamesBrowser_en_US.properties` file (located in the `bin` directory) with the IOR of the Naming Service under the variable `ARTNS`. For example, `ARTNS=IOR:00...`

  When you start the Names Browser and try to connect to a Naming Service there is a checkbox that gives you the option to connect to this Naming Service.

- The `IT_DEFAULT_CLASSPATH` must be altered to include `/opt/iona/lib/OrbixNamesUtils.jar` in it. This is done in the `common.cfg` configuration file.

  Due to an OrbixWeb issue the `orbixweb3.jar` must be added to the configuration:

  ```
  OrbixWeb {
          IT_NAMES_SERVER_HOST="hostname";
          IT_NS_PORT="1570";
          }
  ```

Further Information

For further information about updates to Orbix, including the latest patches, visit the Orbix Update Center at:

```
http://www.iona.com/online/support/update/index.html
```